



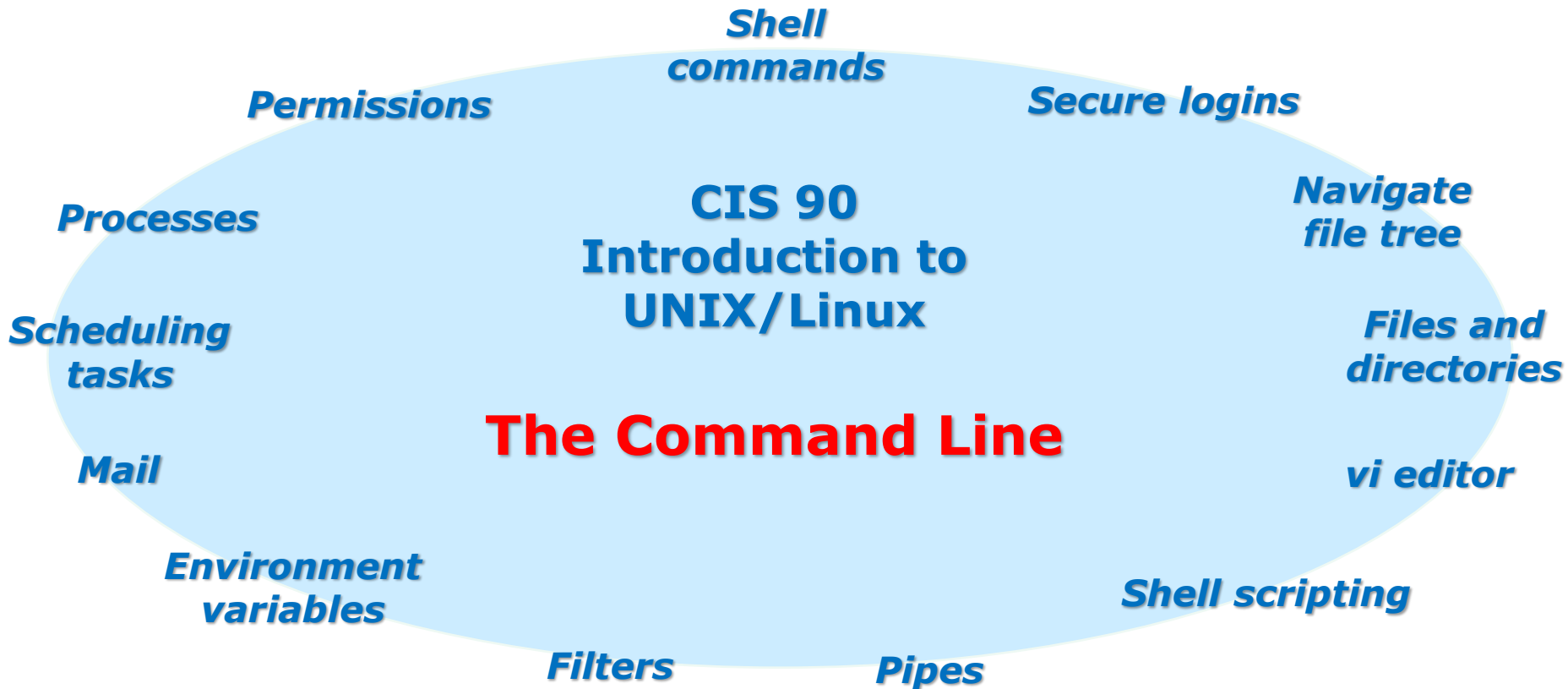
Rich's lesson module checklist

Last Modified 4/29/2018

- ☐ Zoom recording named and published for previous lesson
- ☐ Slides and lab posted
- ☐ Alt WB slides with 1st minute quiz
- ☐ Print out agenda slide and annotate page numbers
- ☐ 1st minute quiz
- ☐ Flash card check
- ☐ Calendar page updated
- ☐ Lab 1 update id-pod-map and test grade script
- ☐ Lab 2 tested (check Q11 kernel release number and finger user account)
- ☐ scripts/schedule-submit-locks
- ☐ Bring Add Codes
- ☐ Bring printed roster
- ☐ Backup slides, whiteboard slides, handouts on flash drive
- ☐ 9V backup battery for microphone
- ☐ Key card for door

☐ <https://zoom.us>

- ☐ Putty + Slides + Chrome
- ☐ Enable/Disable attendee sharing
 - ^ > Advanced Sharing Options > Only Host
- ☐ Enable/Disable attended annotations
 - Share > More > Disable Attendee Sharing



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: <https://web.archive.org/web/20140209023942/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. John's site: <http://teacherjohn.com/>
- Jaclyn Kostner for many webinar best practices: e.g. mug shot page.



Student checklist - Before class starts

The screenshot shows the website simms-teach.com/cis90calendar.php. The page title is "Rich's Cabrillo College CIS Classes CIS 90 Calendar". On the left sidebar, the "CIS 90" link is highlighted. The main content area shows the "CIS 90 (Fall 2014) Calendar" with tabs for "Course Details", "Genders", and "Calendar". The "Calendar" tab is selected, showing a table with columns for "Lesson", "Date", "Topics", and "Link". The first row is for Lesson 1 on 9/2, with topics including "Class and Linux Overview", "Introduction", "Supplemental", "Assignment", "CIS 90 Syllabus", "Quiz 1", and "Commands". Red boxes highlight the "Presentation slides (download)" link, the "Enter virtual classroom" link, and the "CIS 90" link in the sidebar.

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 ConferZoom.
7. Log into Opus-II with Putty or ssh command.



Student checklist - Before class starts

☐ Google

☐ ConferZoom

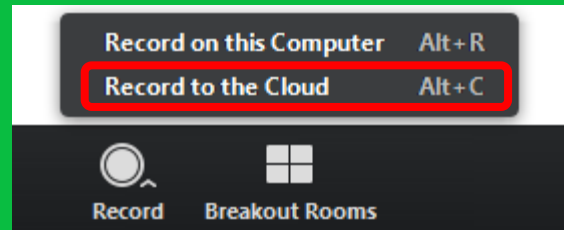
☐ Downloaded PDF of Lesson Slides. I like Foxit Reader so I can take notes using annotations.

The screenshot shows a Zoom meeting interface with several windows open. The main window displays a video player with the text "Get into the car" over a car. Other windows include the Google homepage, the Rich's Cabrillo College CIS 90 website, and a PDF document titled "CIS 90 - Lesson 1". The Zoom toolbar at the bottom shows options like "Unmute", "Start Video", "Invite", "Participants", "Share Screen", "Chat", "Record", and "Leave Meeting".

☐ CIS 90 website Calendar page

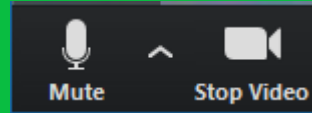
☐ One or more login sessions to Opus-II

Start



Start Recording

Audio Check



Start Recording

Audio & video Check



Instructor: **Rich Simms**
Dial-in: **408-638-0968 (toll)**
Meeting ID: **426 283 384**



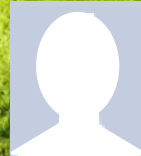
Anita



Shane



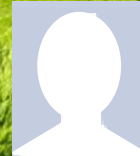
Dan



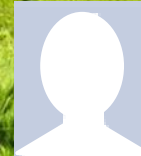
Kage



Justin



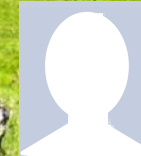
Jo Anne



Moises



Laine



Luis



Christian



Jetta



Cesar



Paul



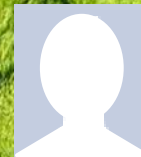
Hilary



Fritz



Jake



Richard



Fergus



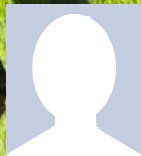
Ciarán



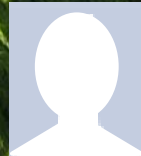
Kimi



November



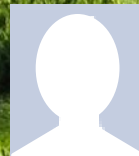
Jaymar



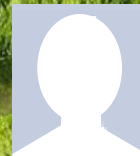
Elena



David



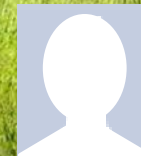
Claudius



Jose



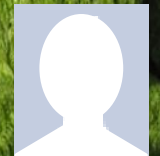
Edgar



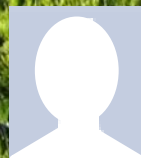
Adam



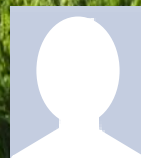
Nathanael T.



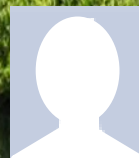
Clara



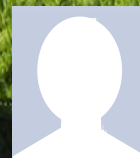
Brandon



Henry



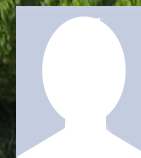
Nate P.



Darren



Nathan K.



David E.

First Minute Quiz

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

Only shown on separate slide deck
at very start of the class

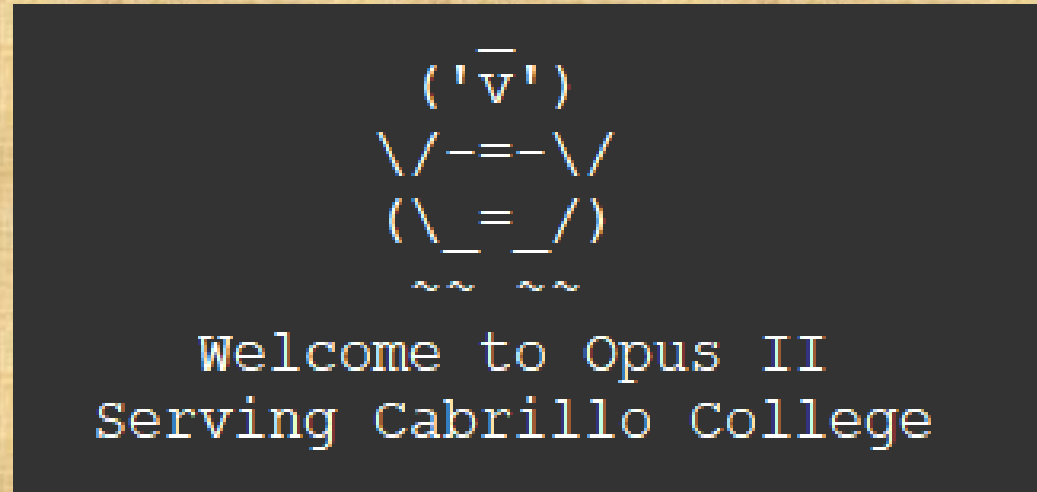
email answers to: risimms@cabrillo.edu

(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



If you haven't already,
log into Opus-II

Class Activity

		Unit 2 Electronic Mail • Guest speaker: Denise Moore on OTC (On-The-Job) training programs • Learn how to use the LINC communication tools: write and /bin/mail • Overview on and to and mail	
3	2/19	Materials • Presentation slides (download)	Lab 2
		Supplemental • Howto #318: Accessing vlab (download)	
		Assignment • Read/skim Lesson 3 slides	

<https://simms-teach.com/cis90calendar.php>

If you haven't already,
download the lesson slides

Class Activity

	• <u>Read/skim Lesson 1 slides</u> • <u>Student Survey</u> • <u>Lab 1</u>	
	ConferZoom • <u>Enter virtual classroom</u> • <u>Class archives</u>	
	Quiz 1 Commenda • Understand how the UNIX login operation	

<https://simms-teach.com/cis90calendar.php>

If you haven't already, join
ConferZoom classroom

Questions



Questions

How this course works?

Past lesson material?

Previous labs?

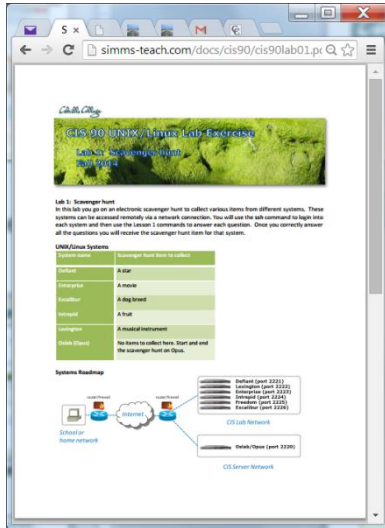
Chinese
Proverb

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.

Lab Assignments -- Pearls of Wisdom

- Don't wait till the last minute to start.
- Plan for things to go wrong and give yourself time to ask questions and get answers.
- 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.
- **Late work is not accepted** so submit what you have for partial credit.



Extra Credit

On the forum

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

On some labs

Extra credit (2 points)

For a small taste of what you would learn in CIS 191 let's add a new user to your Arya VM. Once added we will see how the new account is represented in `/etc/passwd` and `/etc/shadow`.

1. Log into your Arya VM as the cis90 user. Make sure it's your VM and not someone else's.
2. Install the latest updates:
`sudo apt-get update`
`sudo apt-get upgrade`
3. Add a new user account for yourself. You may make whatever username you wish. The example below shows how Benji would make the same username he uses on Opus:
`sudo useradd -G sudo -c "Benji Simms" -m -s /bin/bash simben90`

In lesson slides (search for extra credit)



On the website

<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.

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

• **Website 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 that has been updated but not yet presented in class. (Up to 20 points total)

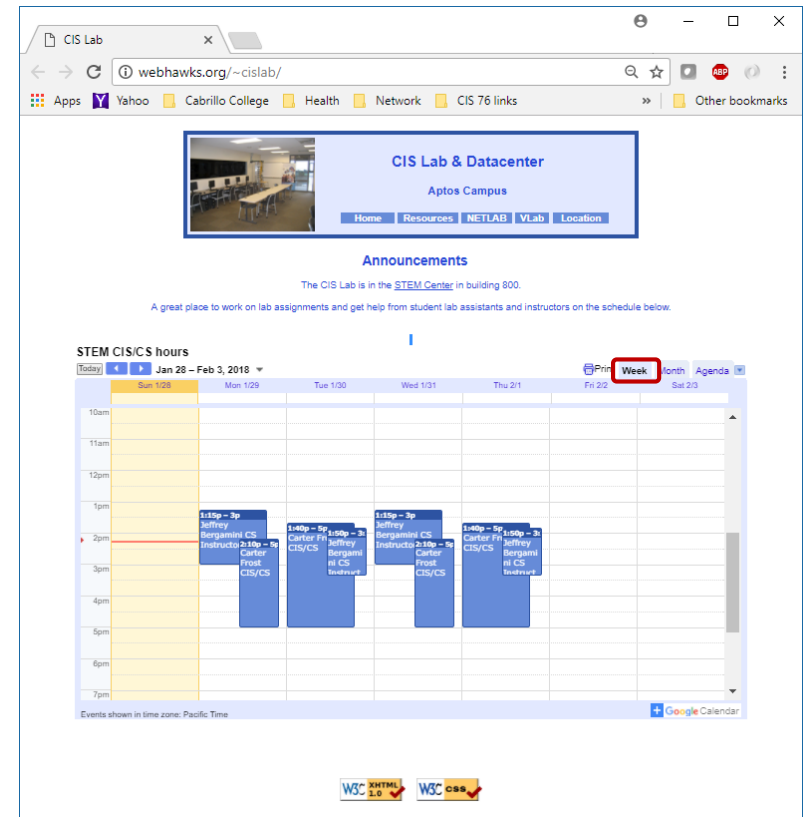
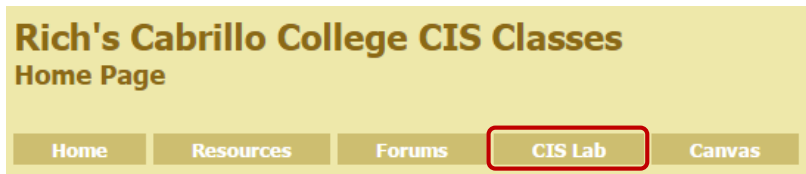
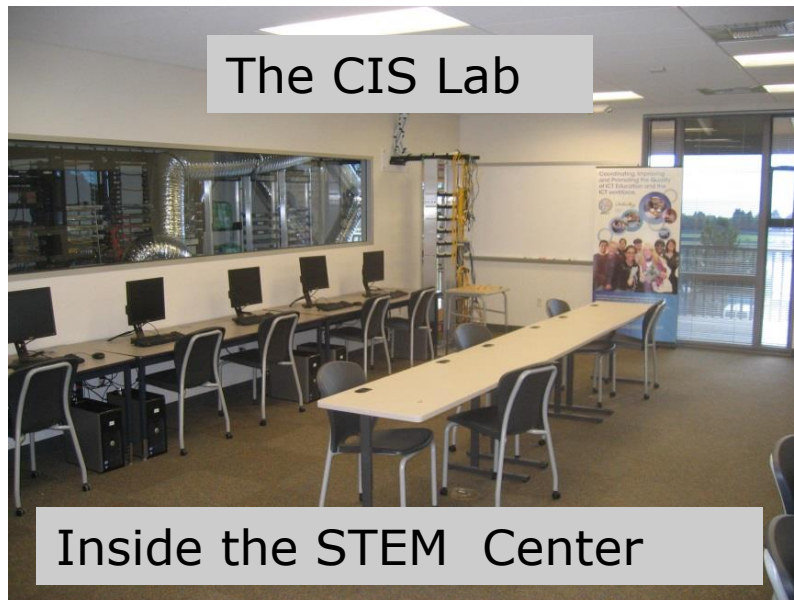
Getting Help When Stuck on a Lab Assignment

- Google the topic/error message.
- Search the Lesson Slides (they are PDFs) for a relevant example on how to do something.
- Post a question on the forum. Explain what you are trying to do and what you have tried so far.
- Talk to a STEM center tutor/assistant.
- Come see me during my office or lab hours. **I will be in the CTC (room 1403) every Wednesday afternoon from 3-5:30.**
- Make use of the Open Questions time at the start of every class.
- Make a cheat sheet of commands and examples so you never again get stuck on the same thing!

Expect to do a LOT of troubleshooting in this course!

Help Available in the CIS Lab

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



To see schedule, click the CIS Lab link on the website and use the "Week" calendar view

CTC - Building 1400 On lower campus



I will be in the CTC (room 1403) every Wednesday
afternoon from 3-5:30



The slippery slope



- 1) If you didn't submit the last lab ...
- 2) If you were in class and didn't submit the last quiz ...
- 3) If you didn't send me the student survey assigned in Lesson 1 ...
- 4) If you haven't made a forum post in the last quarter of the course ...

*Please contact me by email, see me during
my office hours or when I'm in the CTC*

Email: risimms@cabrillo.edu

Logging In (authentication)

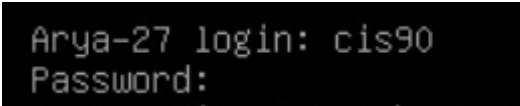


Who goes there?

What's the password?

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

Logging in

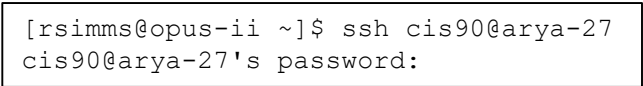
A black rectangular box with white text. The text reads "Arya-27 login: cis90" on the first line and "Password:" on the second line.

```
Arya-27 login: cis90
Password:
```

Virtual terminal login



Graphical desktop login

A terminal window with a white background and black text. It shows a command prompt where the user has entered an SSH command to connect to a host named 'arya-27'. The prompt asks for the password.

```
[rsimms@opus-ii ~]$ ssh cis90@arya-27
cis90@arya-27's password:
```

SSH login

- A system administrator will 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

Where are user accounts and passwords stored?

- User accounts are kept in a file named **/etc/passwd**
- Passwords are kept encrypted in a file named **/etc/shadow**

Note: Systems can also be integrated with a directory service (e.g. Microsoft Active Directory). In that case the user accounts and passwords are will be stored on another server.

The /etc/passwd file

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

```
root:x:0:0:root:/root:/bin/bash
```

The SUPER user is named root

< Snipped >

Regular users

```
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: In spite of its name, 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-II*

```
/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

```
[rsimms@daughter-of-opus ~]$ cat /etc/shadow
cat: /etc/shadow: Permission denied
[rsimms@daughter-of-opus ~]$ sudo cat /etc/shadow
[sudo] password for rsimms:
root:$6$  $  :16226:0:99999:7:::
```

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

The SUPER user is named root

< Snipped >

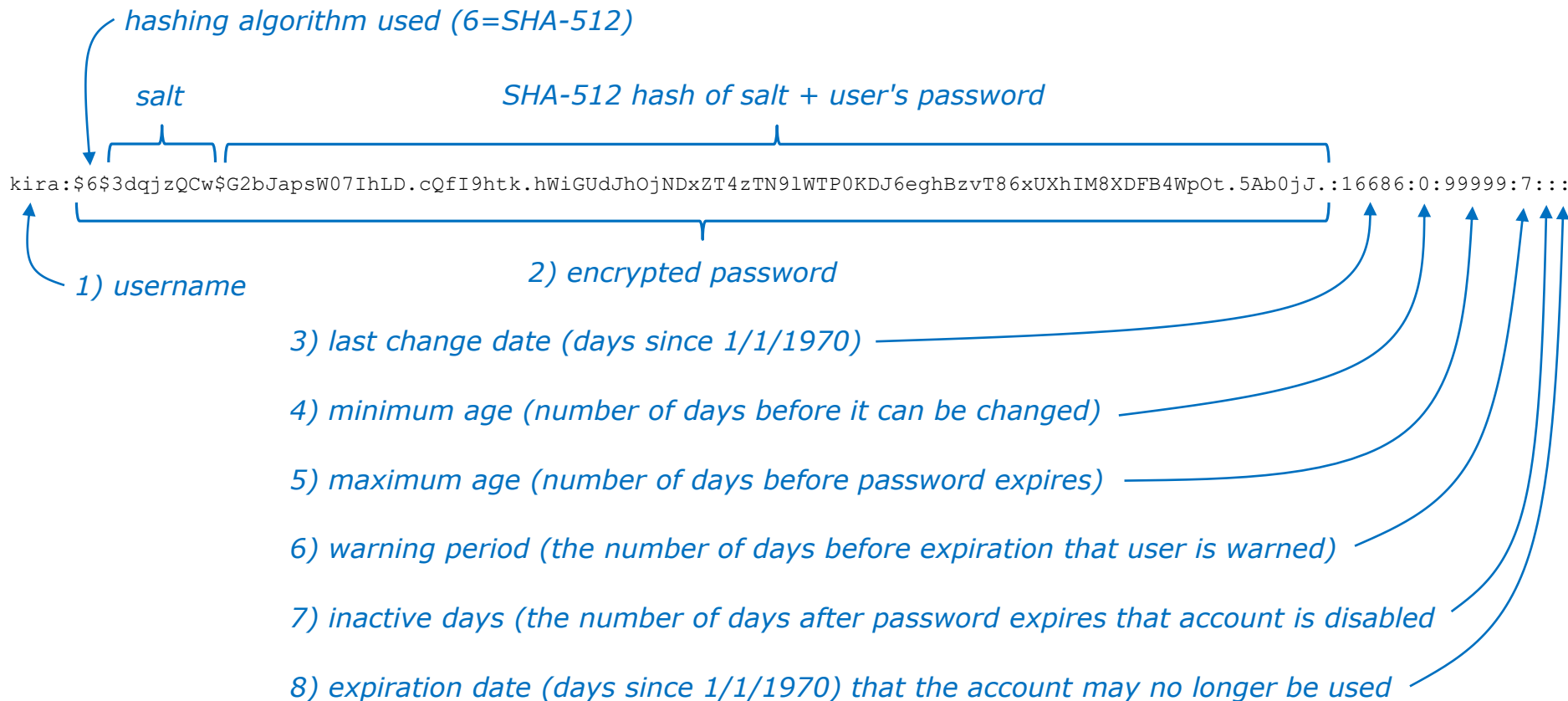
Regular users

```
deanna:$6$hsAXq0Jk$ndIt.oxiFL/qZ7pLAFOaGgxpxAHDEj7ukpd0PfeRN0J9q07Z6Cg0V
3hzo9eSAk0GlaywDtqwL5NefNEEwf9FR1:16686:0:99999:7:::
chakotay:$6$c/kFViIa$ntUJcVJRCut8PwvOSYLlopAI25UsFLNKerGF8OhQIkI78RHTXE1
KOOwvDRSW6BAi4pui7LLpi6JP8QCBMVUls1:16686:0:99999:7:::
kira:$6$3dqjzQCw$G2bJapsW07IhLD.cQfI9htk.hWiGUdJhOjNDxZT4zTN9lWTP0KDJ6eg
hBzvT86xUXhIM8XDFB4WpOt.5Ab0jJ.:16686:0:99999:7:::
chekov:$6$j4PMdv0$HPyW/k04DjMDeLO3qUfEzvQj0fWpLuUWMh9Rv1Ov1V3N/zQxhdhS3
YfSLdhHz0rKBelwzGGx07CrzOfL3MKNa1:16686:0:99999:7:::
[rsimms@daughter-of-opus ~]$
```

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

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

Viewing an account in 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:1201:1090:Benji Simms:/home/cis90/simben:/bin/bash
```

username → *User ID (UID)* → *Group ID (GID)* → *Comment* → *Home directory* → *Shell*
 password (just a placeholder now)

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

Annotate this table with a green check ✓ if you can view this file otherwise a red ✗ if you can't.

Can View	Can't View

For Supplemental Study



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

Excellent article on how passwords created and stored

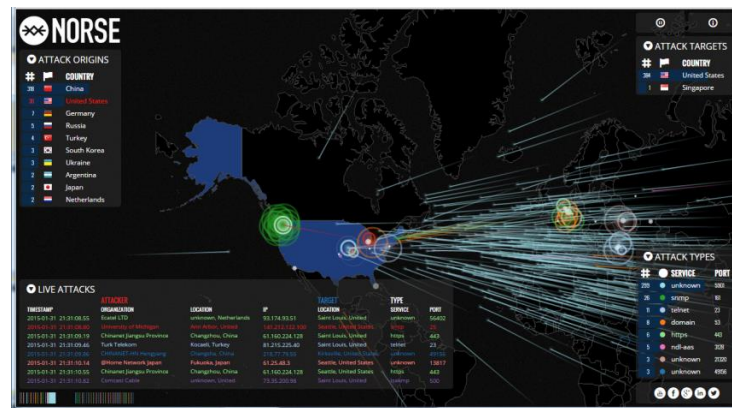
Passwords

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)



<https://www.fireeye.com/cyber-map/threat-map.html>



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

Top source countries

NoSweat : Monday, July 17, 2017

July 17, 2017 – Datacenter is idle over the summer but we still have lots of international visitors!

Source Country	Bytes	Sessions
172.16.0.0-172.31.255.255	6.80 G	352.34 k
192.168.0.0-192.168.255.255	7.80 M	67.41 k
United States	361.73 M	38.05 k
China	159.78 M	22.52 k
Netherlands	33.23 M	8.86 k
France	20.28 M	3.71 k
Ireland	567.29 k	1.39 k
Russian Federation	20.59 M	1.17 k
United Kingdom	334.86 M	776
Nigeria	458.03 M	740
Taiwan ROC	399.90 k	577
Brazil	2.24 M	518
Germany	2.54 M	491
Ukraine	6.62 M	430
Philippines	14.77 M	407
Czech Republic	2.95 M	292
Viet Nam	10.90 M	270
Japan	716.41 k	269
Thailand	5.02 M	264
Belgium	321.48 k	208
India	44.15 M	194
Poland	4.67 M	186
Singapore	319.91 k	170
Hong Kong	1.18 M	166
Greece	95.28 k	163
Sweden	823.54 k	153
Finland	56.20 k	150
Colombia	2.46 M	136

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

*Notice the
brute force
attacks*

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

*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:

188.165.15.181

Check it

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

Report 188.165.15.181

Whois 188.165.15.181

Snoopy - Dashboard

URL: <https://localhost/squert/index.php?id=a45bae5c326b02b0fa95987fd135af9a>

EVENTS SUMMARY VIEWS

INTERVAL: 2015-05-29 00:00:00 -> 2015-05-29 23:59:59 (400:00) FILTERED BY OR

TOGGLE

SUMMARY

queued events	2176
total events	18757
total signatures	33
total sources	-
total destinations	-

COUNT BY PRIORITY

high	66 (3.0%)
medium	-
low	2067 (90.0%)
other	43 (2.0%)

COUNT BY CLASSIFICATION

- compromised L1 -
- compromised L2 -
- attempted access -
- denial of service -
- policy violation -
- reconnaissance -
- malicious -
- no action req'd. 16581 (88.4%)
- escalated event -

HISTORY

QUEUE ACTIVITY LAST EVENT SOURCE COUNTRY DESTINATION COUNTRY

QUEUE	ACTIVITY	LAST EVENT	SOURCE	COUNTRY	DESTINATION	COUNTRY
12		2015-05-29 18:34:25	188.165.15.181	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

ST	TIMESTAMP	EVENT ID	SOURCE	PORT	DESTINATION	PORT	SIGNATURE
RT	2015-05-29 18:27:24	3.8593	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	3.8595	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	3.8598	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	3.8597	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	3.8596	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	3.8595	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	3.8594	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	3.8594	188.165.15.181	56128	207.62.187.230	80	SURICATA STREAM 3way handshake wrong seq wrong ack

WELCOME rsimms | LOGOUT UTC 18:43:54

They never stop trying

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

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

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

Failed logins from:

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

Illegal users from:

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

```
Users logging in through sshd:
guest:
  76.254.22.196 (adsl-76-254-22-196.dsl.pltn13.sbcglobal.net): 2 times
jimg:
  70.132.20.25 (adsl-70-132-20-25.dsl.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

They never stop trying

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

Failed logins from:

122.249.183.95 (x183095.ppp.asahi-net.or.jp): 3 times

218.64.5.131 (131.5.64.218.broad.nc.jx.dynamic.163data.com.cn): 3 times

Illegal users from:

78.46.83.76 (static.76.83.46.78.clients.your-server.de): 3 times

218.4.157.178: 3 times

pam_succeed_if(sshd:auth): error retrieving information about user
teamspeak : 1 time(s)

reverse mapping checking getaddrinfo for
131.5.64.218.broad.nc.jx.dynamic.163data.com.cn failed - POSSIBLE
BREAK-IN ATTEMPT! : 3 time(s)

pam_succeed_if(sshd:auth): error retrieving information about user ts
: 2 time(s)

pam_succeed_if(sshd:auth): error retrieving information about user
plcmspip : 2 time(s)

pam_succeed_if(sshd:auth): error retrieving information about user
PlcmSpIp : 1 time(s)

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

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

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

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

Wh01e#!!!!	(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:

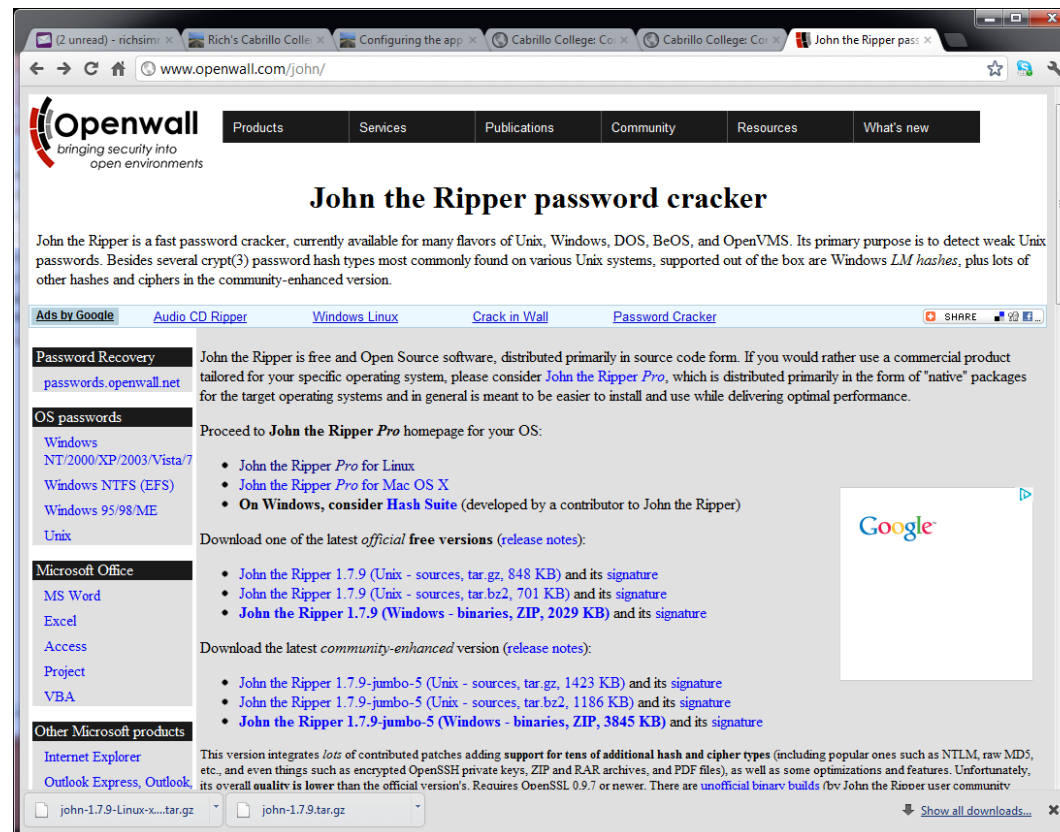
```
/home/cis90/simmsben $ passwd  
Changing password for user simben90.  
Changing password for simben90  
(current) UNIX password:   
New UNIX password:   
Retype new UNIX password:   
passwd: all authentication tokens updated successfully.  
/home/cis90/simmsben $
```

*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

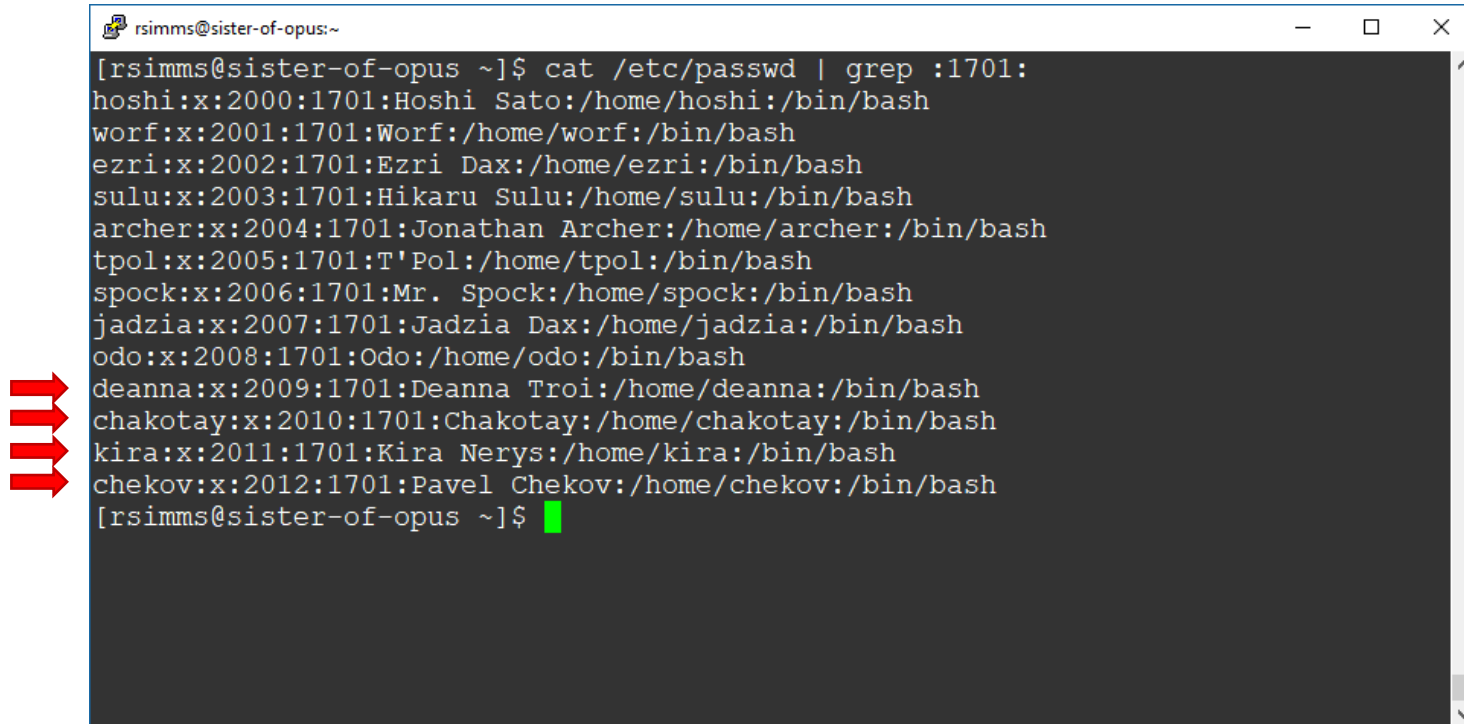
An open source cracker that tries common passwords first followed by a brute force dictionary attack



Instructor:

Use sister-of-opus and john* aliases to demo.
Show password.1st for common passwords.

/etc/passwd



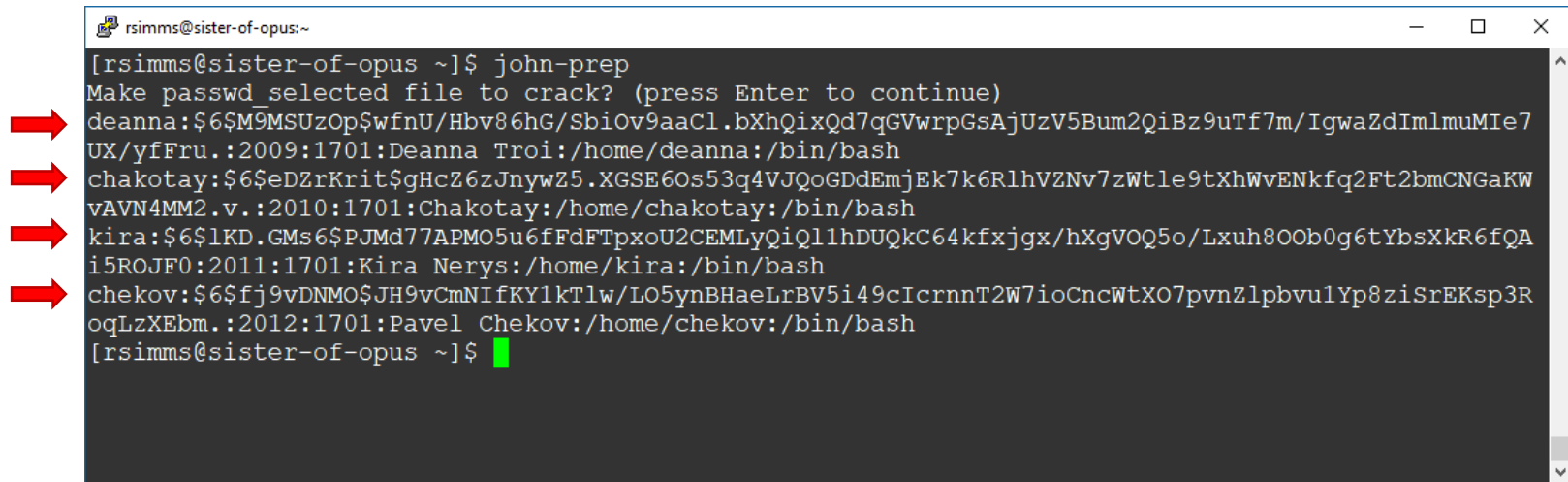
```
rsimms@sister-of-opus:~  
[rsimms@sister-of-opus ~]$ cat /etc/passwd | grep :1701:  
hoshi:x:2000:1701:Hoshi Sato:/home/hoshi:/bin/bash  
worf:x:2001:1701:Worf:/home/worf:/bin/bash  
ezri:x:2002:1701:Ezri Dax:/home/ezri:/bin/bash  
sulu:x:2003:1701:Hikaru Sulu:/home/sulu:/bin/bash  
archer:x:2004:1701:Jonathan Archer:/home/archer:/bin/bash  
tpol:x:2005:1701:T'Pol:/home/tpol:/bin/bash  
spock:x:2006:1701:Mr. Spock:/home/spock:/bin/bash  
jadzia:x:2007:1701:Jadzia Dax:/home/jadzia:/bin/bash  
odo:x:2008:1701:Odo:/home/odo:/bin/bash  
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@sister-of-opus ~]$
```

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

1234567
secret
terces
chekov1

/etc/shadow

john-prep

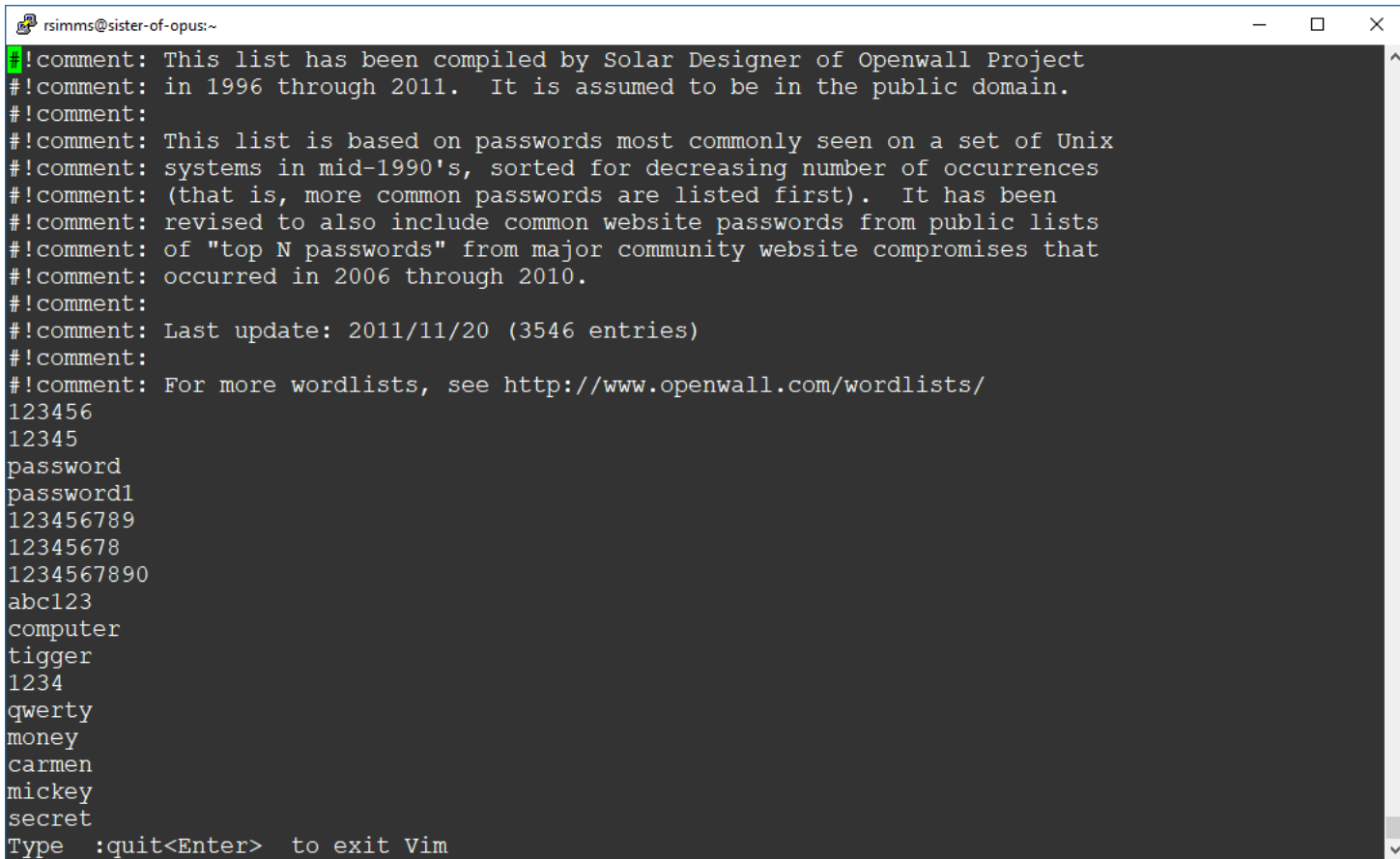


```
[rsimms@sister-of-opus ~]$ john-prep
Make passwd selected file to crack? (press Enter to continue)
deanna:$6$M9MSUzOp$wfnU/Hbv86hG/SbiOv9aaCl.bXhQixQd7qGVwrpGsAjUzV5Bum2QiBz9uTf7m/IgwaZdImlmuMie7
UX/yfFru.:2009:1701:Deanna Troi:/home/deanna:/bin/bash
chakotay:$6$eDZrKrit$gHcZ6zJnywZ5.XGSE6Os53q4VJQoGDdEmjEk7k6RlhVZNv7zWtle9tXhWvENkfq2Ft2bmCNGaKW
vAVN4MM2.v.:2010:1701:Chakotay:/home/chakotay:/bin/bash
kira:$6$lKD.GMs6$PJMd77APMO5u6fFdFTpxoU2CEMLyQiQl1hDUQkC64kfxjgx/hXgVOQ5o/Lxuh80Ob0g6tYbsXkR6fQA
i5ROJF0:2011:1701:Kira Nerys:/home/kira:/bin/bash
chekov:$6$fj9vDNMO$JH9vCmNifKY1kTlw/LO5ynBHaeLrBV5i49cIcrnnT2W7ioCncWtXO7pvnZlpbvulYp8ziSrEKsp3R
oqLzXEbm.:2012:1701:Pavel Chekov:/home/chekov:/bin/bash
[rsimms@sister-of-opus ~]$
```

*Encrypted (hashed) passwords in /etc/shadow for
deanna, chakotay, kira and chekov*

password.1st

view security/john-1.8.0-jumbo-1/run/password.1st



```
rsimms@sister-of-opus:~  
#!/comment: This list has been compiled by Solar Designer of Openwall Project  
#!/comment: in 1996 through 2011. It is assumed to be in the public domain.  
#!/comment:  
#!/comment: This list is based on passwords most commonly seen on a set of Unix  
#!/comment: systems in mid-1990's, sorted for decreasing number of occurrences  
#!/comment: (that is, more common passwords are listed first). It has been  
#!/comment: revised to also include common website passwords from public lists  
#!/comment: of "top N passwords" from major community website compromises that  
#!/comment: occurred in 2006 through 2010.  
#!/comment:  
#!/comment: Last update: 2011/11/20 (3546 entries)  
#!/comment:  
#!/comment: For more wordlists, see http://www.openwall.com/wordlists/  
123456  
12345  
password  
password1  
123456789  
12345678  
1234567890  
abc123  
computer  
tigger  
1234  
qwerty  
money  
carmen  
mickey  
secret  
Type :quit<Enter> to exit Vim
```

Common passwords John will try first

Cracking their passwords

john-run

```
rsimms@sister-of-opus:~  
[rsimms@sister-of-opus ~]$ john-run  
Start cracking passwords? (press Enter to continue)  
  
Mon Feb  5 16:23:36 PST 2018  
  
Warning: detected hash type "sha512crypt", but the string is also recognized as "crypt"  
Use the "--format=crypt" option to force loading these as that type instead  
Loaded 4 password hashes with 4 different salts (sha512crypt, crypt(3) $6$ [SHA512 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)  
terces           (kira)  
4g 0:00:02:00 DONE 2/3 (2018-02-05 16:25) 0.03313g/s 302.7p/s 306.7c/s 306.7C/s retupmoc..dlanod  
Use the "--show" option to display all of the cracked passwords reliably  
Session completed  
  
Mon Feb  5 16:25:37 PST 2018  
  
[rsimms@sister-of-opus ~]$
```

Doesn't take very long but these are very weak passwords!

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 1/2 hour and a half minute explanation of the Password Haystacks concept. Click this link to view their quick introduction.

GRC's Interactive Brute Force Password "Search Space" Calculator
(WARNING: you do have your own "brute force" attack! What happens then, stays there.)

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

dummy

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

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

Time Required to Exhaustively Search this Password's Space:	
Online Attack Scenario: (Assuming one thousand guesses per second)	3.43 hours
Offline Fast Attack Scenario: (Assuming one hundred billion guesses per second)	0.000124 seconds
Massive Cracking Array Scenario: (Assuming one hundred trillion guesses per second)	0.00000124 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

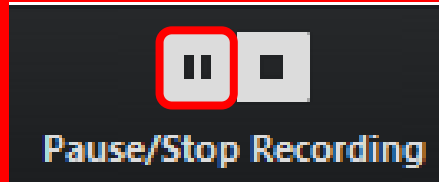
Grading Rubric (30 points)

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 with Refund is this Saturday.
4. Guest speaker next week on internships & apprenticeships.



Pause Recording

Audio Check

Roll Call

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

risimms@cabrillo.edu



Resume/Stop Recording

Resume Recording

Audio Check

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				
dwalin	Grade				
elrond	Grade				
eomer	Grade				
galadriel	Grade				
gimli	Grade				
glorfindel	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 rsimms@cabrillo.edu)

*See Lesson 1
assignments on
the Calendar*

To get notifications of new forum posts

Subscribe to the forum to get email notifications of new posts

After logging in:

1. Go to the CIS 90 class forum.
2. At the bottom of the page, click the "Subscribe forum" link on the lower left. When subscribed you get email notifications when new posts are made.
3. To unsubscribe, click it again.

[Home](#) < [Board index](#) ☒ [Subscribe forum](#)

*Unsubscribed
looks like this.*

[Home](#) < [Board index](#) ☐ [Unsubscribe forum](#)

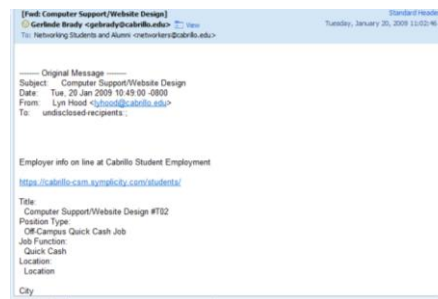
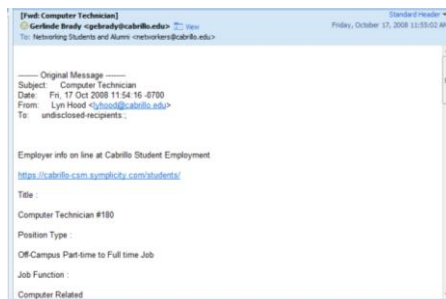
*Subscribed
looks like this.*

Cabrillo Networking Program Mailing list

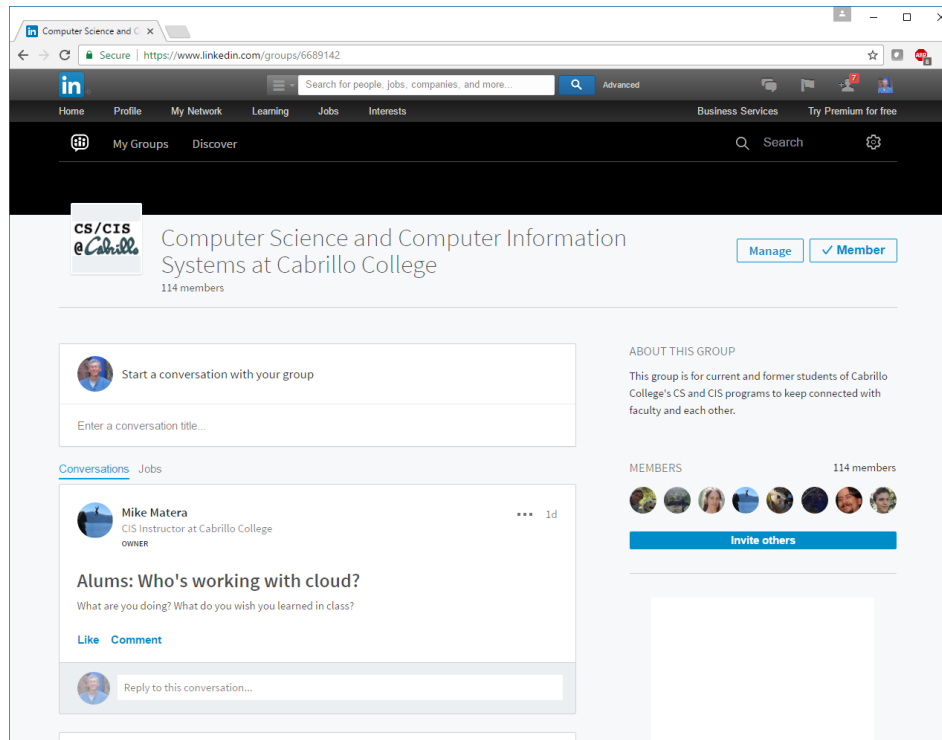
Subscribe by sending an email (no subject or body) to:

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

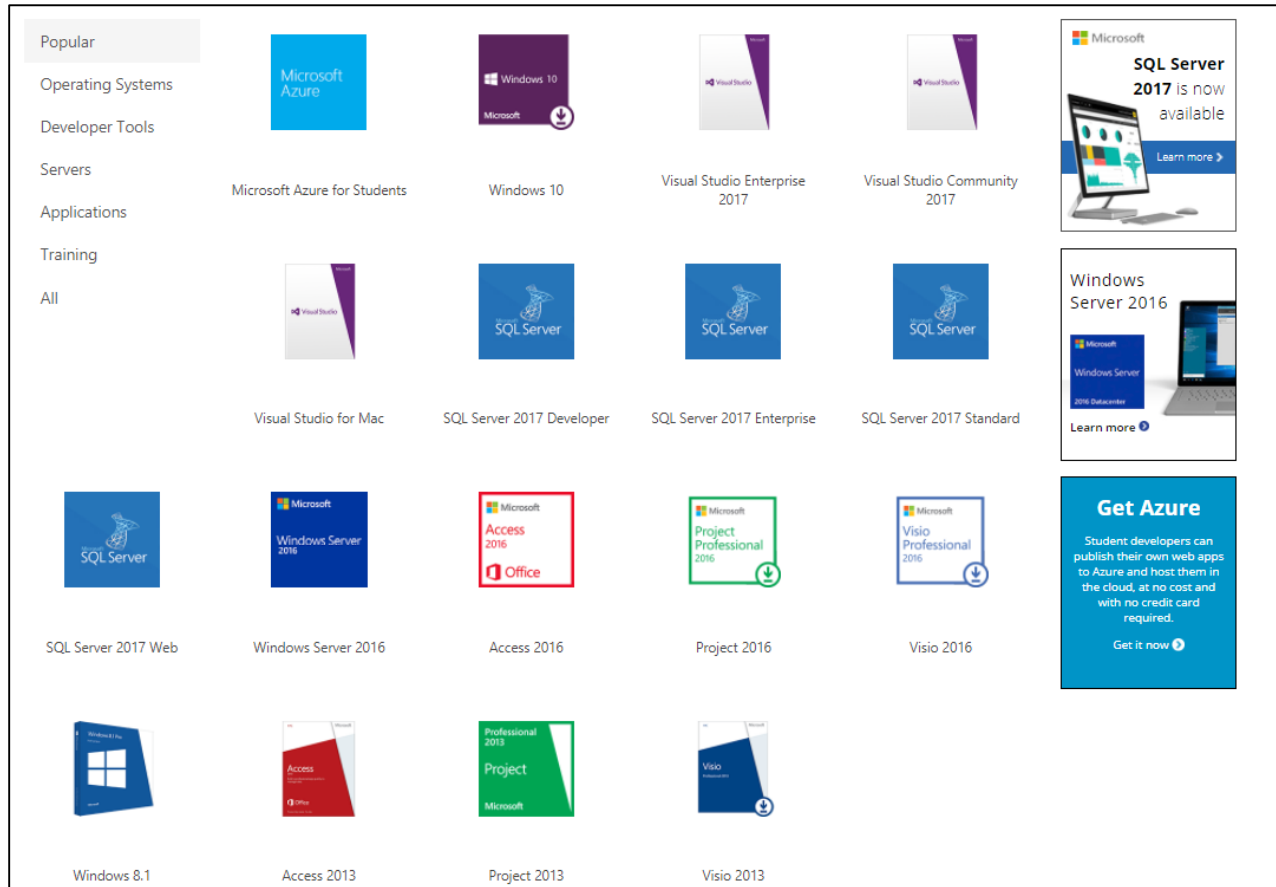


For 3 points extra credit:

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

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

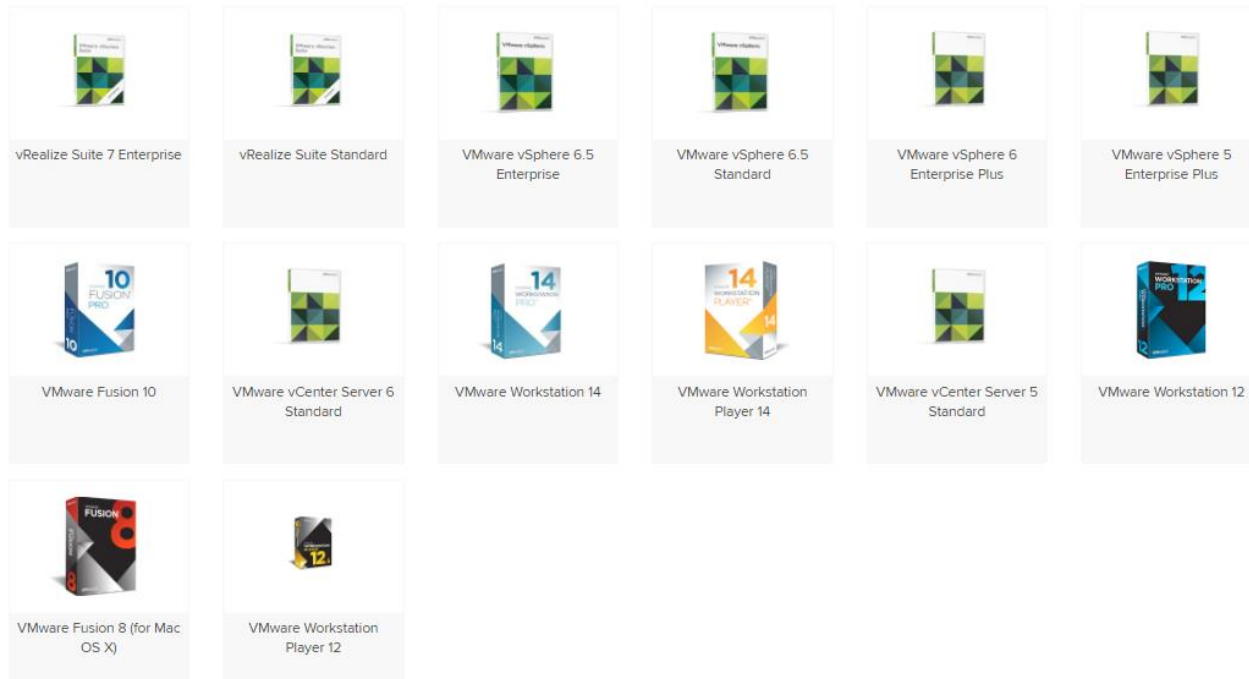
Microsoft Academic Webstore



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

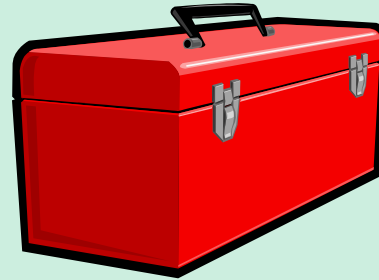
To get to this page, go to **<http://simms-teach.com/resources>** and click on the appropriate link in the *Academic Software for CIS Students* section.

VMware Academic Webstore



- 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 *Academic Software for CIS Students* 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

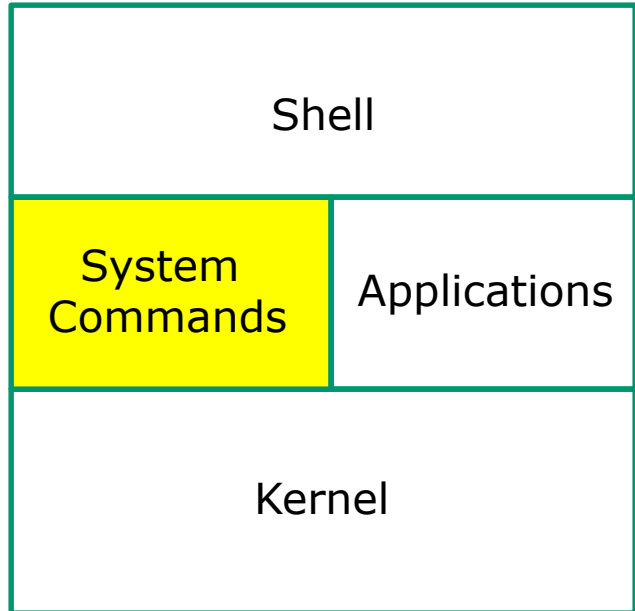
- List all shell variables

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-ii ~]$ 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@opus-ii ~]$ type bogus
```

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

bogus is not on the user's path

```
[rsimms@opus-ii ~]$ 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@opus-ii ~]$ 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)
```

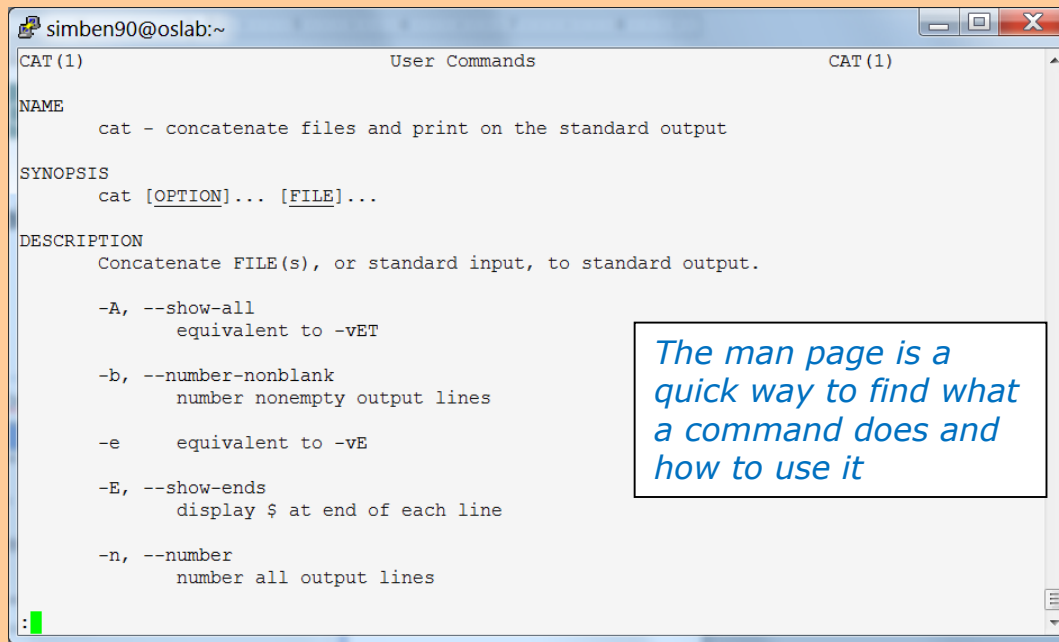
man command

Show the manual page (documentation) for a command

Syntax:

man *command*

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

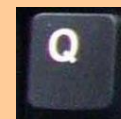


The screenshot shows a terminal window titled 'simben90@oslab:~'. The terminal displays the output of the 'man cat' command. The output is formatted with sections: NAME, SYNOPSIS, DESCRIPTION, and a list of options. The options listed are: -A, --show-all (equivalent to -vET), -b, --number-nonblank (number nonempty output lines), -e (equivalent to -vE), -E, --show-ends (display \$ at end of each line), and -n, --number (number all output lines). A text box is overlaid on the right side of the terminal window, containing the text: 'The man page is a quick way to find what a command does and how to use it'.

```
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  
  
:
```



*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'.

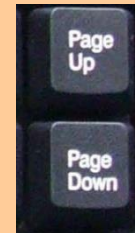
    scale = 20

    /* 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

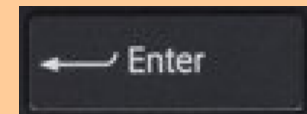
```



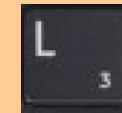
Use these keys to scroll



Use q key to quit



Use Enter to follow a link ()*



Use L to go back to last page

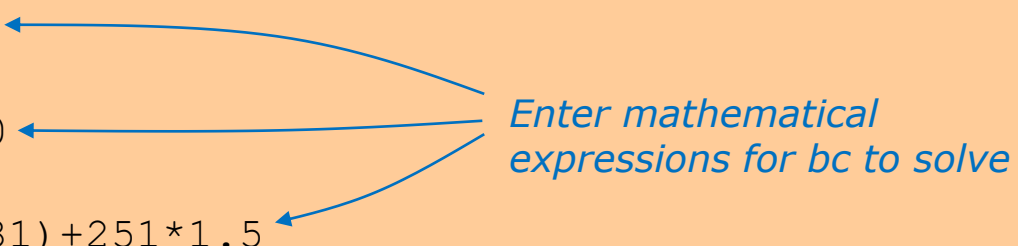
*Move cursor over an * and press Enter to follow link*

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*

Class Activity

1) Is **red** a UNIX command?

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

2) Is **blue** a UNIX command?

Type your answers in the chat window

```
root@kali:~# whatis red
red (1) - red color
root@kali:~# man red
red (1) - red color
root@kali:~# whatis blue
blue (1) - blue color
root@kali:~#
```

Class Activity

1) What does the following mathematical expression reduce to?

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

Type your answer in the chat window

The Shell Path

The Path

The shell uses your path to locate commands to execute

- A path is an 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 is not on your path the shell will print an error message like:

-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**.
- To locate a command on your path use: **type *command*** where *command* is the name of the command you want to locate.

Show your shell path

```
/home/cis90/simben $ echo $PATH  
/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/home/cis90/si  
mben/../../bin:/home/cis90/simben/bin:.
```

The colon character is used to separate directories on the path

1st directory: /usr/local/bin
2nd directory: /usr/bin
3rd directory: /usr/local/sbin
4th directory: /usr/sbin
5th directory: /home/cis90/simben/../../bin
6th directory: /home/cis90/simben/bin
7th directory: .



Locations of common commands

/bin

```

[raimma@server0-01 ~]$ cat /etc/passwd
arch:mut:fgrep:ls:pwd:sync
ash:hsh:date:awk:mail:red:tar
aw:ash:static:dd:grep:mail:tr:trsh
awk:df:gtar:mkmod:radir:touch
username:dmwag:qunip:mktmp:ipm:true
hsh:hsh:domainname:cpio:rvl:uname
hsh2:hsh2:doxex:hovtame:mount:rvlview:uname
hsh:hsh:domainname:lgawk:mt:sed:unlockd_start
aw:aw:dummykey:ls-ls-ls:sv:editont:unlockd_stop
shgrrp:scho:kill:node:netstat:sterial:unlik
mkmod:ed:kill:alice:sh:usleep
boom:boom:grep:link:mkdomainname:sleep:vi
apple:apple:mv:ls:psawk:sort:view
iplo:iplo:cat:loadkeys:ping:stty:yfdomainname
hsh:hsh:dummy:ps:sa:zcat
[raimma@server0-01 ~]$

```

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

/usr/bin

[illegible]

/sbin

```
[root@server0-01 ~]# rpm -qa | grep rsync
```

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

/usr/sbin

```

[isimms@server-01 ~]$ isisms % ls /usr/sbin
accept          ntpd
adduser          nupdate
addl-connect    nupdate
addl-setup      ntp-generate
addl-start      ntpd
addl-status     ntpdate
addl-stop       ntpdmonst
alternatives    ntptrace
amavisd          ntp-wait
analog          ntpdnav
arping          packer
atd             ncbltestl
atun            nlog
atunconf        nmap_dump
autocomb        nmap_wst
avmcapictrl    rpd
cronos-activation-sysconf  rpdump
build-locale-archive  rppoc
build-index-control  rppoc-relay
csmel-lock-helper  rppoc-server
csmplint        rppoc-smiff
cshfontpath     rpsmets
cshfontpath     prliases

```

*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***

Red Hat 7 and Centos 7

```
/home/cis90/simben $ ls -l /bin /sbin
lrwxrwxrwx. 1 root root 7 Aug  4 2017 /bin -> usr/bin
lrwxrwxrwx. 1 root root 8 Aug  4 2017 /sbin -> usr/sbin
```

Starting with Red Hat and Centos version 7 the /bin and /usr/bin commands have been combined into the /usr/bin directory.

Same with /sbin and /usr/sbin.

Heads up on future tests

Memorize these five directories:

```
/bin  
/usr/bin  
/sbin  
/usr/sbin  
/usr/local/bin
```

I will mess with your path on almost every test!

To fix things you will need to look at your path and insure it has these directories!

Locate a command on the path

Example: Where is the **cal** command located?

```
[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*

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

```
/usr/local/bin:/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/home/cis90/simben/../../bin:/home/cis90/simben/bin:..
```

```
/home/cis90/simben $ ls -l /bin /sbin
```

*The **/usr/bin** directory is third directory on the path*

Locate a command on the path

Example: Where is the **bogus** command located?

```
/home/cis90/simben $ type bogus
```

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

*there is no command named bogus
in any of the path directories*

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

```
/usr/local/bin:/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/home/cis90/simben/../../bin:/home/cis90/simben/bin:.
```

```
/home/cis90/simben $ ls -l /bin /sbin
```

Locate a command on the path

Example: Where is the **type** command located?

```
/home/cis90/simben $ type type  
type is a shell builtin
```

the type command is built into the shell

```
/home/cis90/simben $ echo $PATH  
/usr/local/bin:/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/home/cis90/simben/../../bin:/home/cis90/simben/bin:.  
/home/cis90/simben $ ls -l /bin /sbin
```

Class Activity

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

echo

/usr/bin

route

Built into the shell

scavenge

/usr/sbin

submit

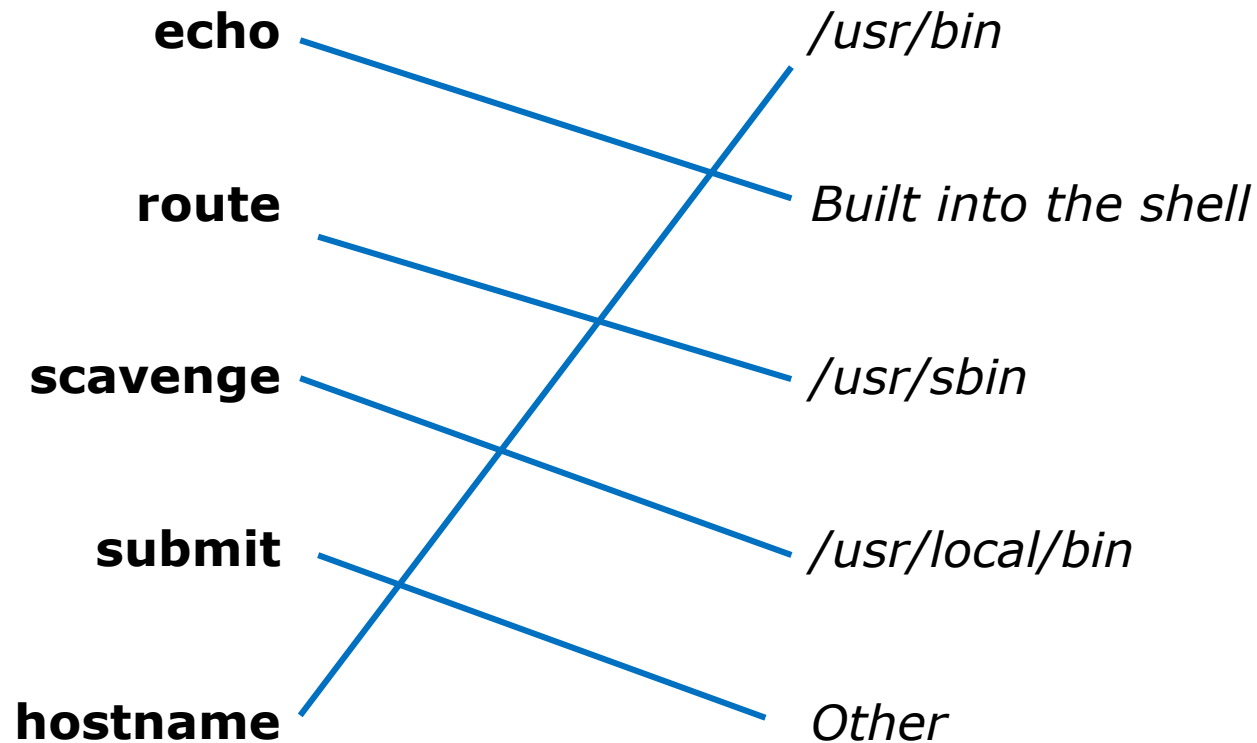
/usr/local/bin

hostname

Other

Class Activity

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





Programs

Binary code
vs text scripts



UNIX commands & utilities are executable programs

A program can be binary code:

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

Class Activity

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

Hint: use the **type** command with hostname as the argument.

Type your answer in the chat window.

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

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

Type your answer in the chat window.

3) Can you **cat** the **hostname** command?

Paste a line of output in the chat window.

4) Is **hostname** a UNIX command?

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

Type your answer in the chat window.

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.

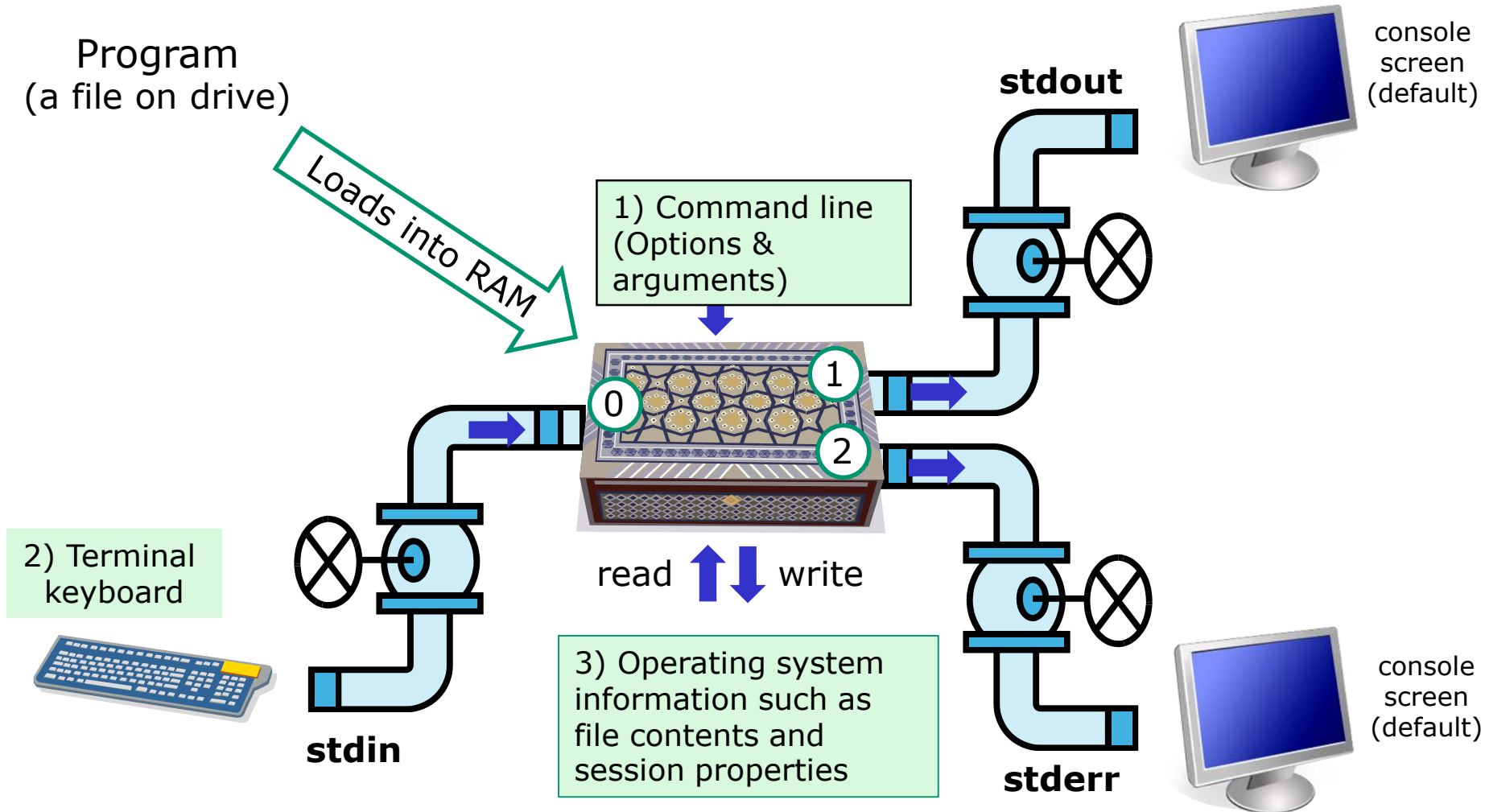
Type your answer in the chat window.

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?

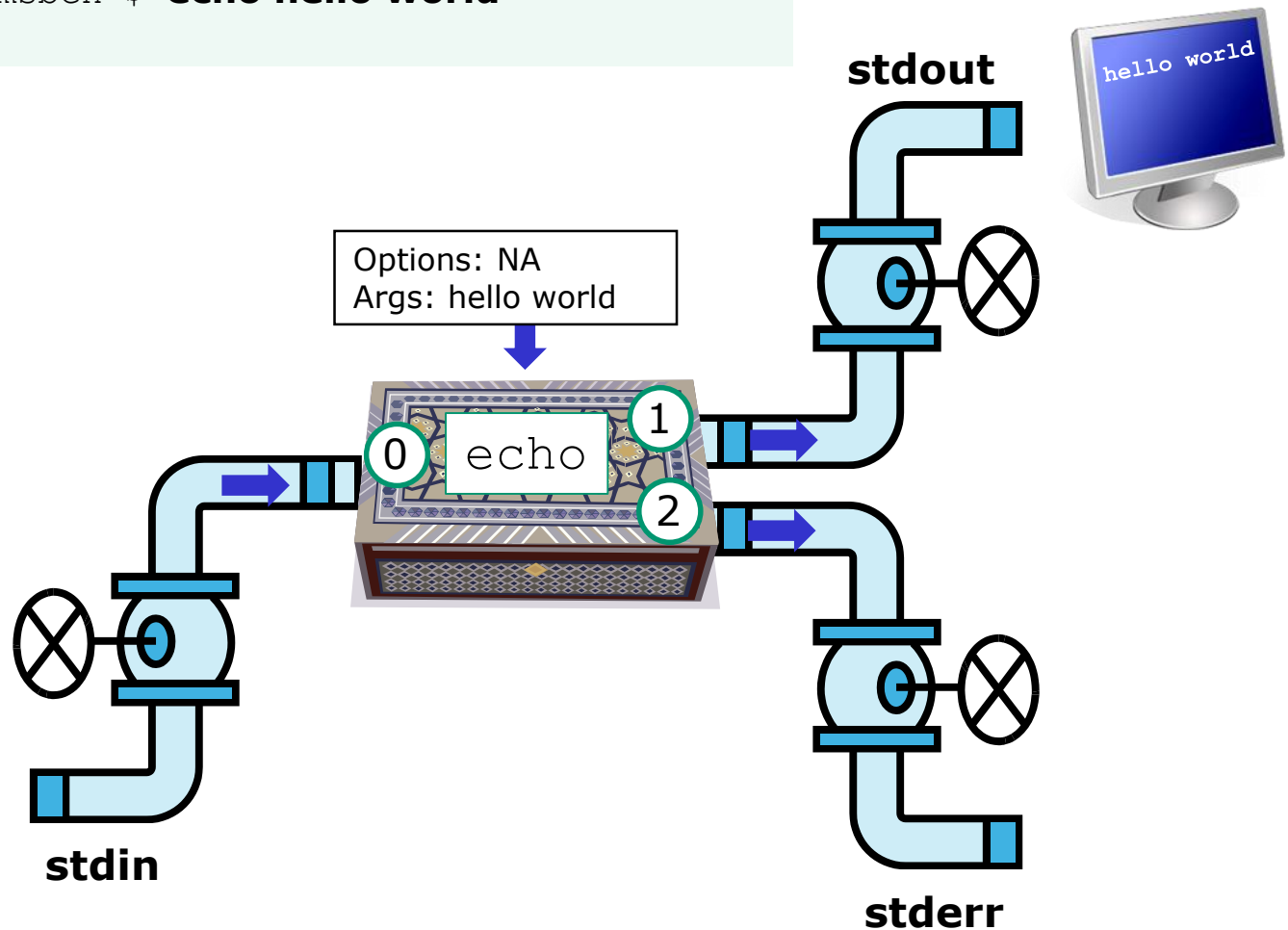
```
/home/cis90/simmen $ echo hello world  
hello world
```

```
/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?

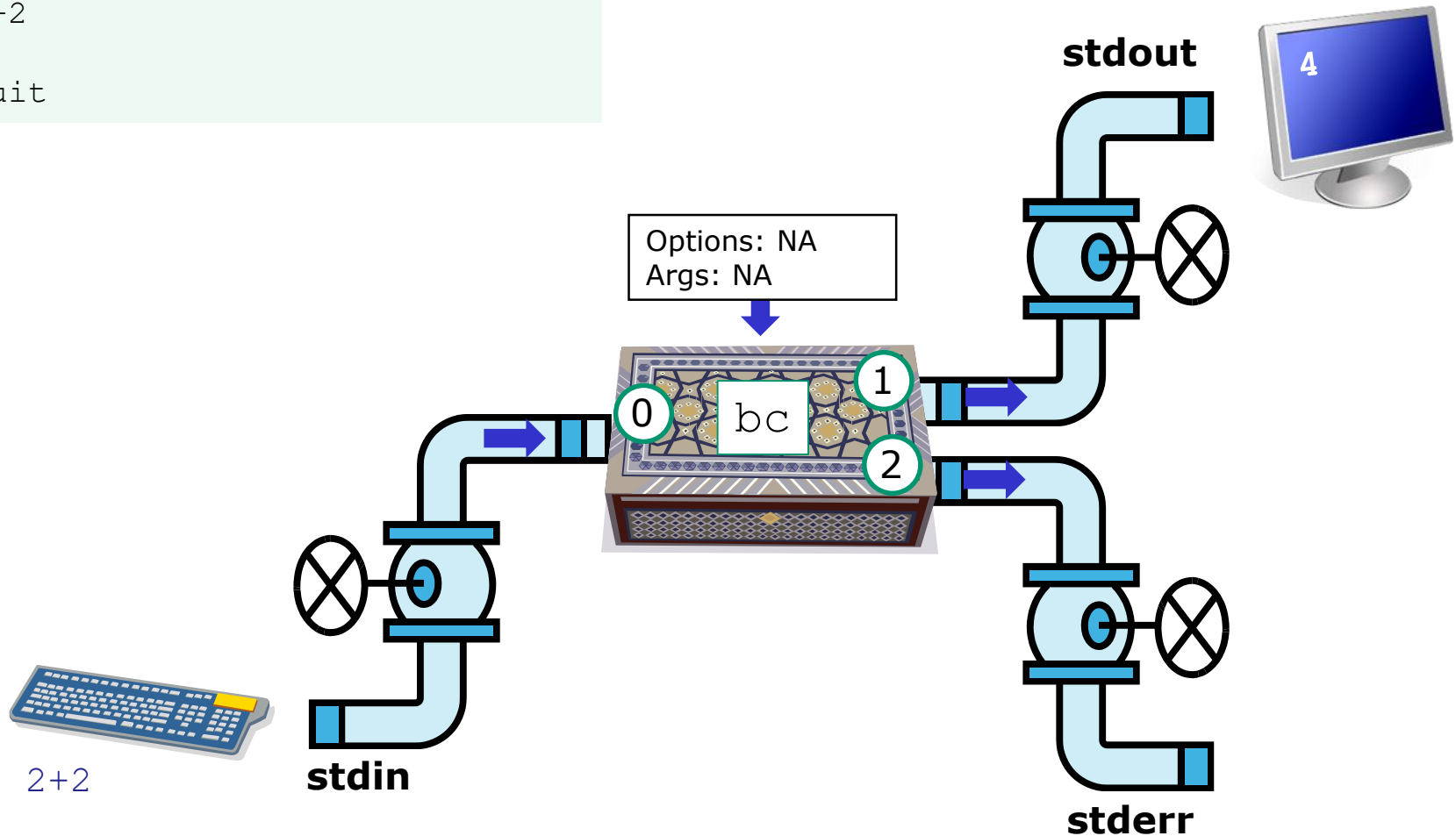
```
/home/cis90/simmsben $ bc
bc 1.06
Copyright 1991-1994, 1997, 1998, 2000 Free
Software Foundation, Inc.
This is free software with ABSOLUTELY NO
WARRANTY.
For details type `warranty'.
2+2
4
500-200+3
303
sqrt(64)
8
quit
```

```
/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)
```

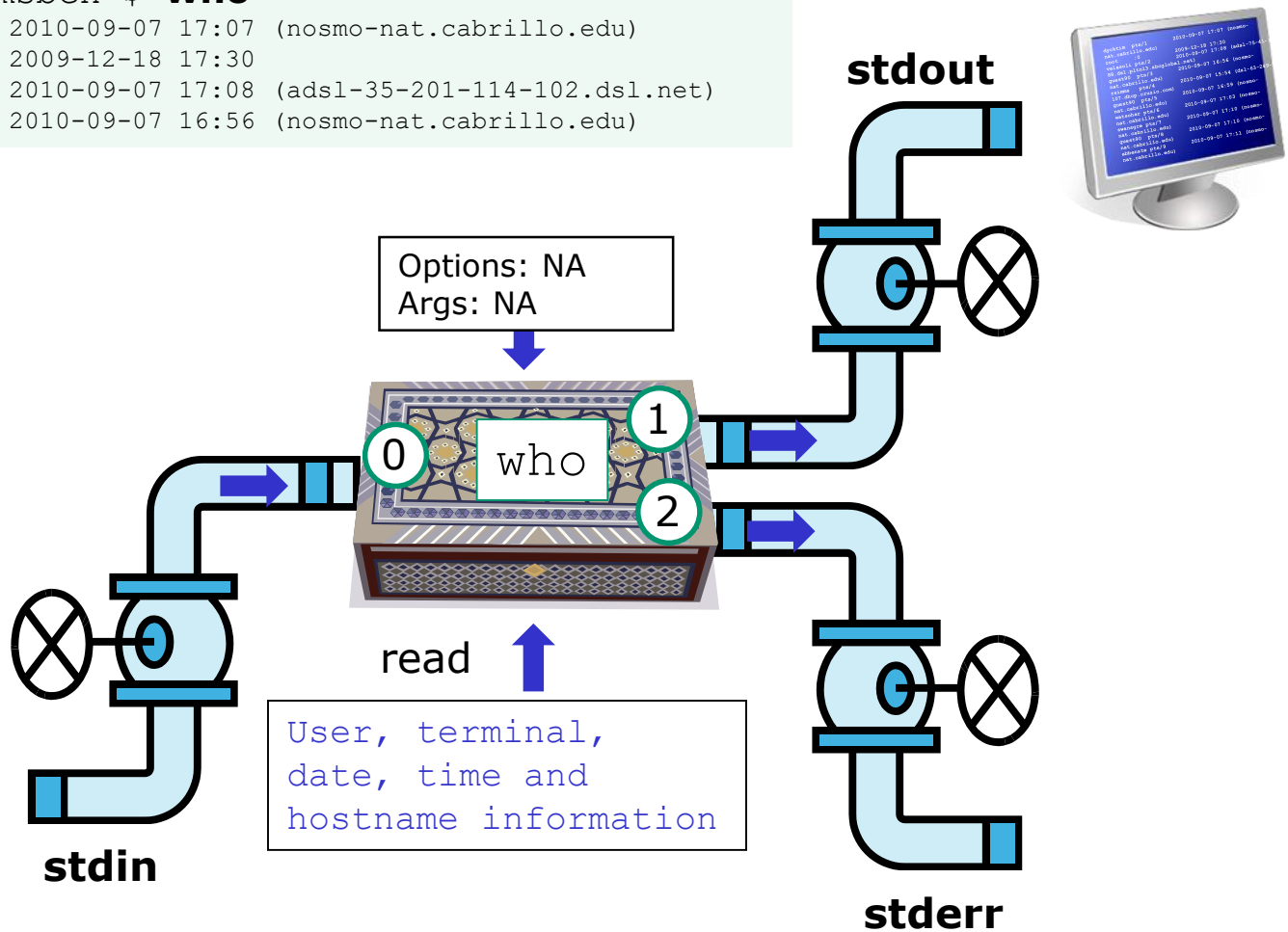
```
/home/cis90/simben $ uname
Linux
```

*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

Is this **ps** command getting its input from the command line, the keyboard or the operating system?

```
/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`

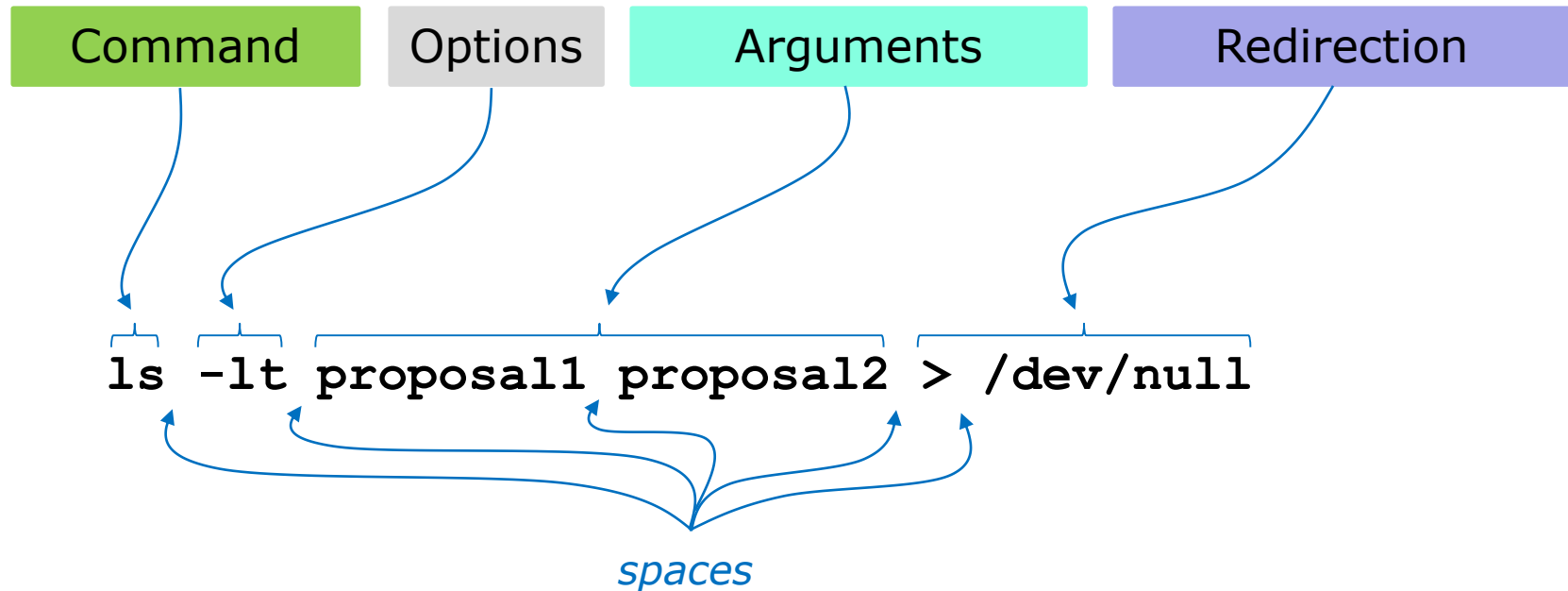
Arguments – follow the options. Multiple arguments must be separated by spaces.

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
```

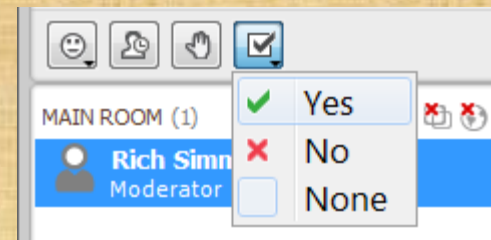
```
What can I do for you simben90?
```

```
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@opus-ii ~]$ date
```

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

```
[simben90@opus-ii ~]$
```

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

user name

hostname

indicates regular user

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

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 Variables set command

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

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

```
LS_COLORS='rs=0:di=01;34:ln=01;36:mh=00:pi=40;33:so=01;35:do=01;35:bd=40;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:*.lзма=01;31:*.tlz=01;31:*.txz=01;31:*.zip=01;31:*.z=01;31:*.Z=01;31:*.dz=01;31:*.gz=01;31:*.lz=01;31:*.xz=01;31:*.bz2=01;31:*.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:'
MACHTYPE=i386-redhat-linux-gnu
MAIL=/var/spool/mail/simben90
MAILCHECK=60
OLDPWD=/bin
OPTERR=1
OPTIND=1
OSTYPE=linux-gnu
PATH=/usr/lib/qt-3.3/bin:/usr/local/bin:/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/sbin:/home/cis90/simben/./bin:/home/cis90/simben/bin:.
PIPESTATUS=([0]="127")
PPID=17309
PROMPT_COMMAND='printf "\033]0;%s@%s:%s\007" "${USER}" "${HOSTNAME%%.*}" "${PWD/#$HOME/~}"'
PS1='$PWD $ '
PS2='> '
PS4='+ '
PWD=/home/cis90/simben
QTDIR=/usr/lib/qt-3.3
QTINC=/usr/lib/qt-3.3/include
QTLIB=/usr/lib/qt-3.3/lib
SELINUX_LEVEL_REQUESTED=
SELINUX_ROLE_REQUESTED=
SELINUX_USE_CURRENT_RANGE=
SHELL=/bin/bash
SHELLOPTS=braceexpand:emacs:hashall:histexpand:history:ignoreeof:interactive-comments:monitor
SHLV=1
SSH_CLIENT='50.0.68.235 51849 2220'
SSH_CONNECTION='50.0.68.235 51849 172.30.5.20 2220'
SSH_TTY=/dev/pts/2
TERM=xterm
UID=1001
USER=simben90
USERNAME=
_=ser
colors=/etc/DIR_COLORS
/home/cis90/simben $
```

Shell (Environment) Variables

env command

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

```
HOSTNAME=opus-ii.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)*



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

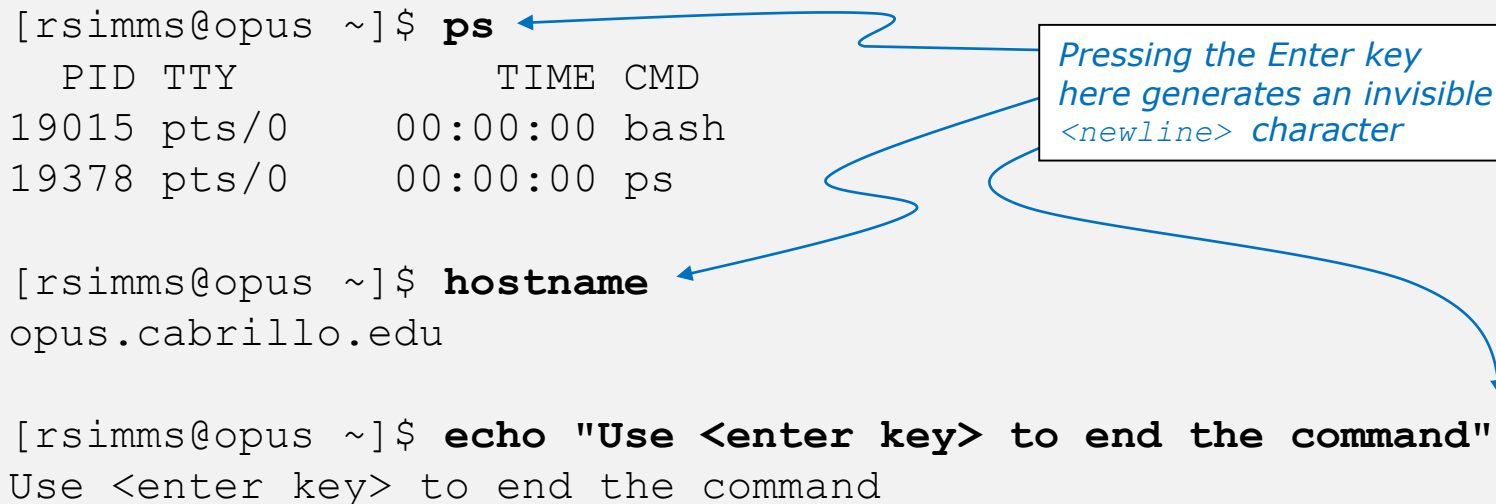
```
/home/cis90/simben $ echo '"double quotes"'  
"double quotes"
```

```
/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. Blue arrows point from a text box to the end of each command line, indicating that pressing the Enter key at that point generates a 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@opus-ii ~]$ echo a b c d e f
a b c d e f
```

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

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

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

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

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

Escape the double quote marks

Metacharacters - ; (semi-colon)

The semi-colon ; allows multiple commands on one line

```
[simmsben@opus-ii 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/
opus-ii.cis.cabrillo.edu
-bash: name: command not found
simben90
jerusalem tiger
/home/cis90/simben $ hostname; uname; echo $LOGNAME; ls Poems/Blake/
opus-ii.cis.cabrillo.edu
Linux
simben90
jerusalem tiger
  
```

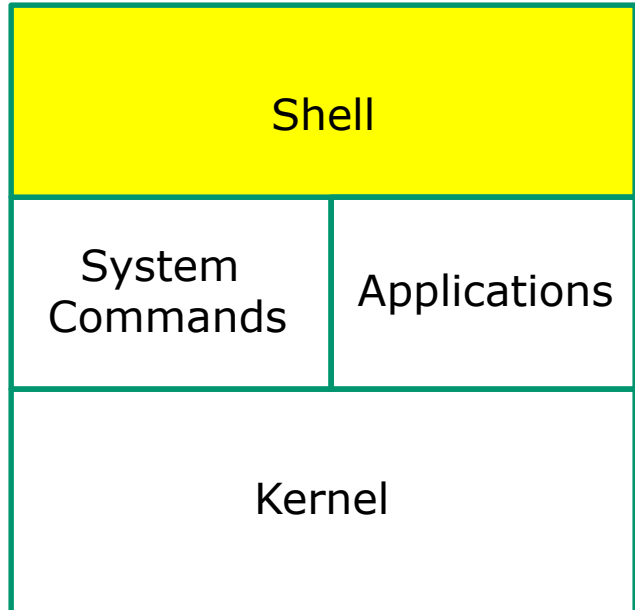
Press <tab> after the P and B and the shell fills in the rest

Press up arrow and the shell retypes the previous command

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

The Shell (six steps)

The Shell

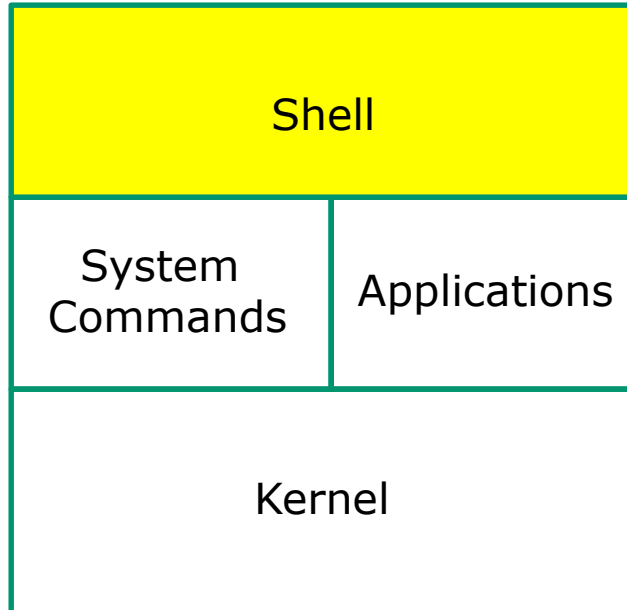


- Allows users to interact with the computer via a “**command line**”.
- **Prompts** for a command, parses the command, finds the right program and gets that program executed.
- Is called a “**shell**” because it hides the underlying operating system.
- Multiple shell programs are available: **sh** (Bourne 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:

<i>prompt</i>	<i>command</i>	
/home/cis90/simben \$	ls -lt proposal1 proposal2	
-rw-r--r--.	1 simben90 cis90 1074 Aug 26 2003 proposal1	} <i>output</i>
-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/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/home/cis90/simben/../../bin:
/home/cis90/simben/bin:.
```

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

```
1st directory: /usr/local/bin    nope, not found here
2nd directory: /usr/bin        bingo, found here!
3rd directory: /usr/local/sbin
4th directory: /usr/sbin
5th directory: /home/cis90/simben/../../bin
6th directory: /home/cis90/simben/bin
7th directory: .
```

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

Try mimicking what the shell does to search for ls:

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

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



Life of the Shell

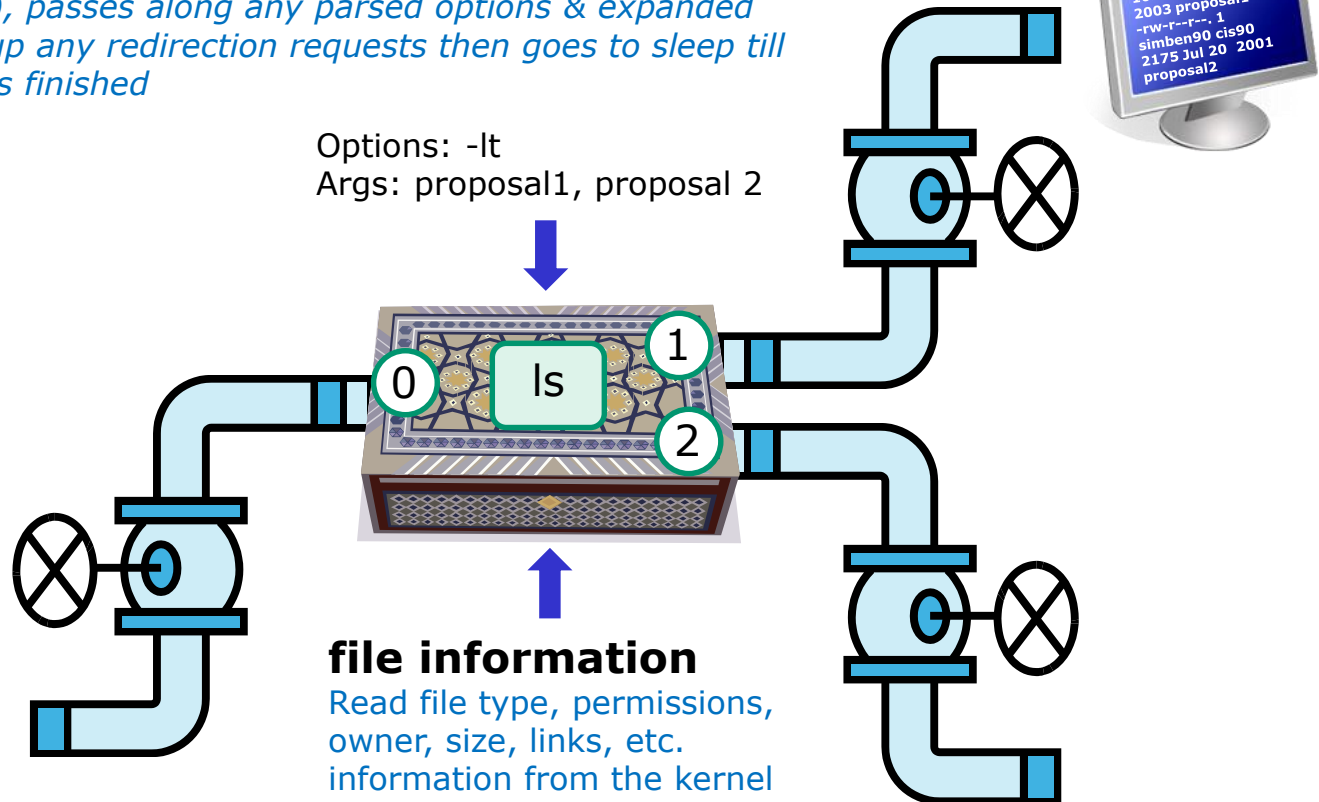
Shell Steps

- 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**

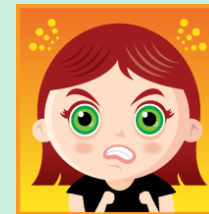
-bash: ls-lt: command not found

What's wrong?

Who output the error?

Life without a path

-bash: xxxx: command not found



Don't get mad, just fix your path!

Life without a path

<https://simms-teach.com/docs/cis90/cis90-life-with-no-path.pdf>



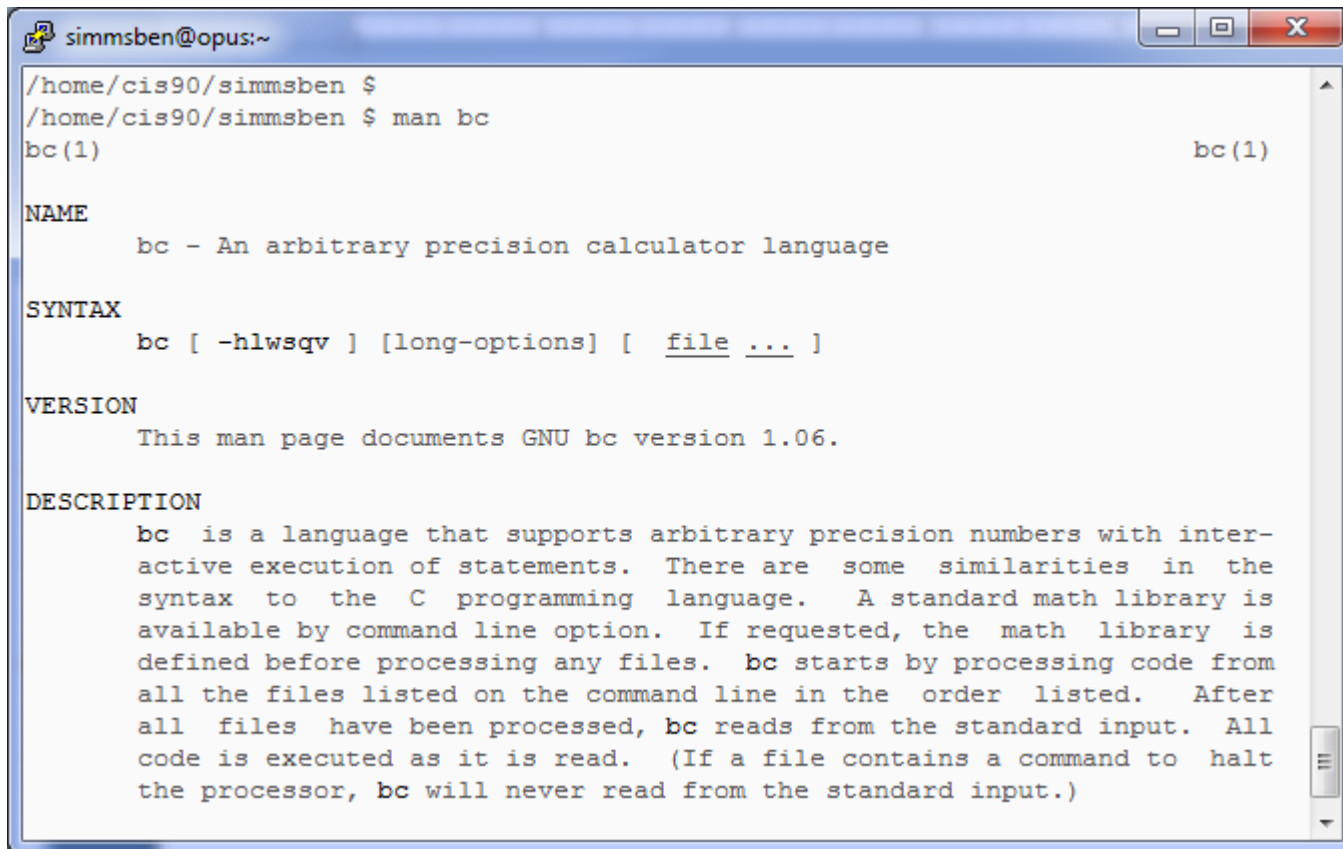
*Need a fresh start -- just log out
and back in again and your path
will be back to normal!*

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 displays a Google search for "linux bc command", showing approximately 1,180,000 results. The foreground window shows the "linux.about.com" page for the "bc" command. The page includes a PayPal advertisement, a search bar, 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 with a math library.

Google Search Results:

- Search query: linux bc command
- About 1,180,000 results (0.24 seconds)
- Results include:
 - bc - Linux Command - Unix
 - Linux / Unix Command Library: b
 - examples.
 - linux.about.com/od/commands/l
 - Linux and UNIX bc comman
 - linking you to information about the
 - www.computerhope.com/unix/ubc
 - command-line calculations u
 - bc is included with (almost?) all Li
 - math library functions in the bc co
 - www.basicallytech.com/blog/index
 - Command line calculator, bc
 - How to do calculation if I only have
 - very complicated calculation. To pe
 - linux.byexamples.com/archives/...
 - Linux bc Command- Basic
 - What is Linux bc Command? ...
 - above command displays the sum
 - www.hscripts.com/tutorials/linux-4
 - bc: A Handy Utility | Linux Jo
 - Mr. McAndrew shows us how the l
 - algorithms. Linux, as with almost
 - www.linuxjournal.com/article/2544

linux.about.com: Linux

Home > Computing & Technology > Linux

Advertisement: PayPal merchants are virtually immune to identity theft and scammers. DID YOU KNOW... LEARN MORE

Search: linux bc command

Linux / Unix Command: bc

Command Library

Sponsored Links:

- Google Apps for Business
- California Online Classes
- Unix Commands Tutorial

NAME

bc - An arbitrary precision calculator language

SYNTAX

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

DESCRIPTION

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

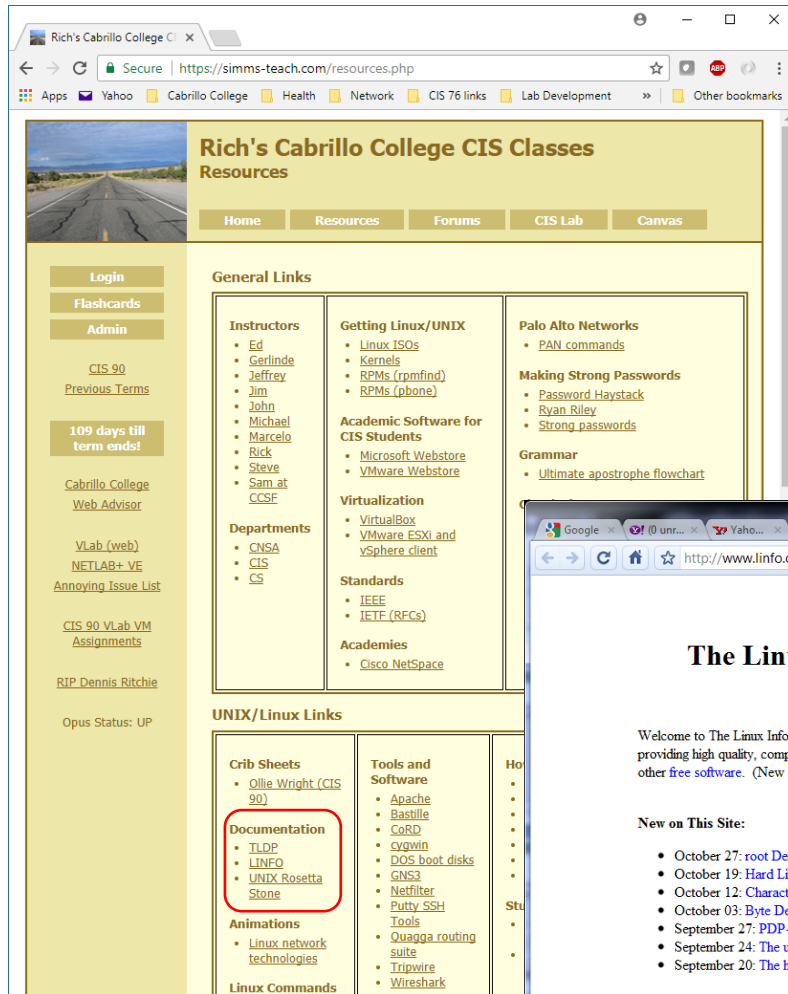
Advertisement: PayPal MERCHANT SERVICES. DID YOU KNOW... The world's largest retailers use PayPal payment processing. Walmart, lenovo, eToys.com

Other Documentation

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

Documentation

Two of my favorite documentation links



Rich's Cabrillo College CIS Classes Resources

Home Resources Forums CIS Lab Canvas

Login
Flashcards
Admin

CIS 90
Previous Terms

109 days till term ends!

Cabrillo College
Web Advisor

VLab (web)
NETLAB+ VE
Annoying Issue List

CIS 90 VLab VM
Assignments

RIP Dennis Ritchie

Opus Status: UP

General Links

Instructors

- Ed
- Gerlinde
- Jeffrey
- Jim
- John
- Michael
- Marcelo
- Rick
- Steve
- Sam at CCSF

Getting Linux/UNIX

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

Academic Software for CIS Students

- Microsoft Webstore
- VMware Webstore

Virtualization

- VirtualBox
- VMware ESXi and vSphere client

Departments

- CNSA
- CIS
- CS

Standards

- IEEE
- IETF (RFCs)

Academies

- Cisco NetSpace

Palo Alto Networks

- PAN commands

Making Strong Passwords

- Password Haystack
- Ryan Riley
- Strong passwords

Grammar

- Ultimate apostrophe flowchart

UNIX/Linux Links

Crib Sheets

- Ollie Wright (CIS 90)

Documentation

- TLDP
- LINFO
- UNIX Rosetta Stone

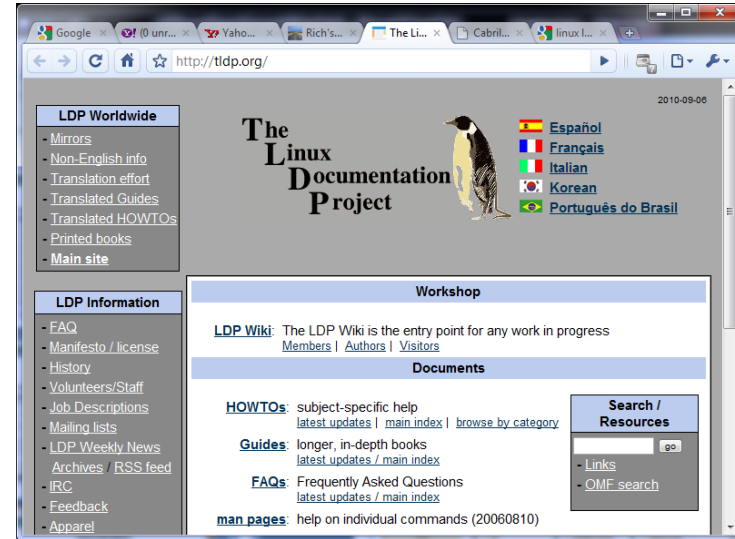
Animations

- Linux network technologies

Linux Commands

Tools and Software

- Apache
- Bastille
- CoRD
- CVS
- DOS boot disks
- GNS3
- Netfilter
- Putty SSH
- Tools
- Quagga routing suite
- Tripwire
- Wireshark



http://tldp.org/

2010-09-06

The Linux Documentation Project

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

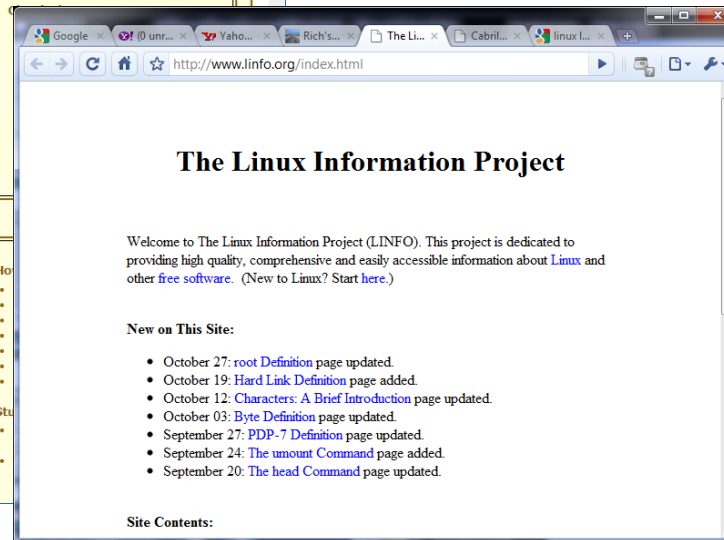
Guides: longer, in-depth books
latest updates / main index

FAQs: Frequently Asked Questions
latest updates / main index

man pages: help on individual commands (20060810)

Search / Resources

- Links
- OMF search



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

The Linux Information Project

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

New on This Site:

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

Site Contents:

The Linux Documentation and Information Projects

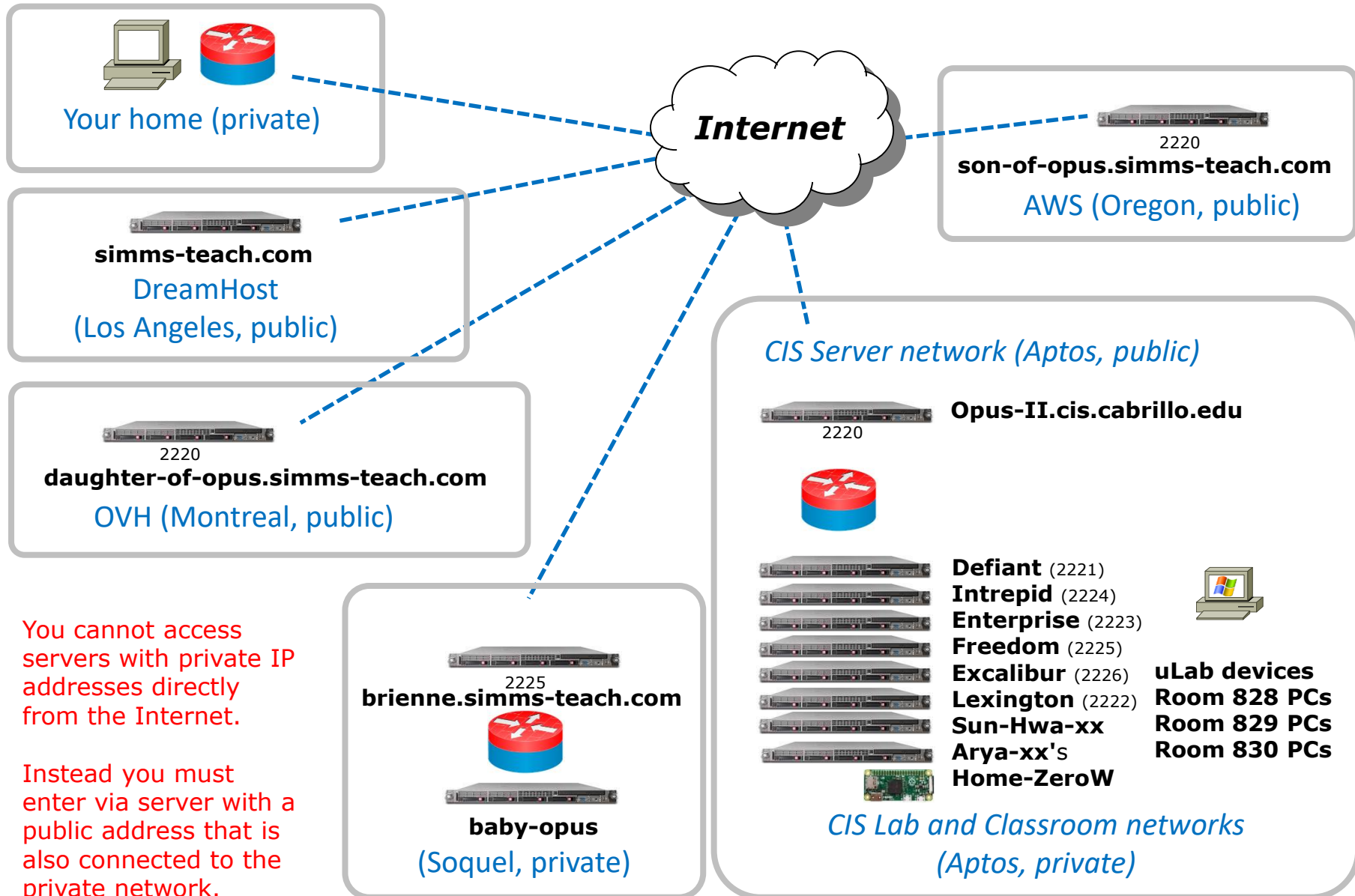


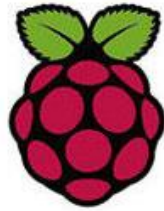
If we have time,
otherwise cover
ZeroW, VLab, Virtual
Terminal modules to
Lesson 3



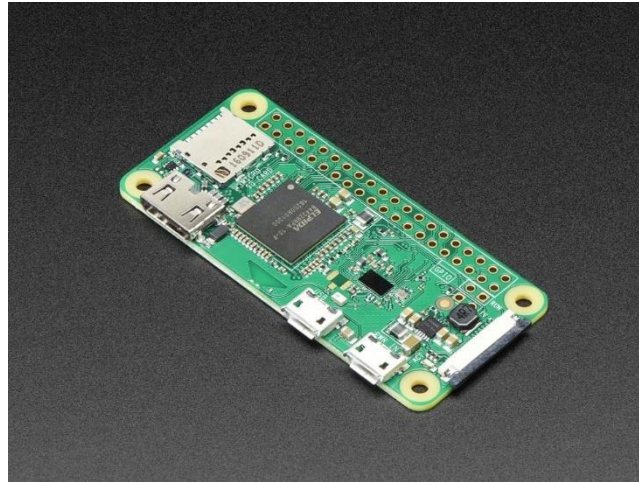
Navigating the Internet using SSH

Second driving lesson continued





Raspberry Pi



Raspberry Pi Zero W

<https://www.adafruit.com/products/3400>

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

Rich's Cabrillo College CIS Classes Home Page

Home Resources Forums CIS Lab Canvas

Login
Flashcards
Admin

CIS 90
[Previous Terms](#)

1 day till term starts!

Cabrillo College
[Web Advisor](#)

VLab (web)
NETLAB+ VE
[Annoying Issue List](#)

CIS 90 VLab VM Assignments

[RIP Dennis Ritchie](#)

Opus Status: UP

Rich Simms

Contact

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

My Spring 2018 Cabrillo Classes

- CIS 90 - Introduction to UNIX/Linux

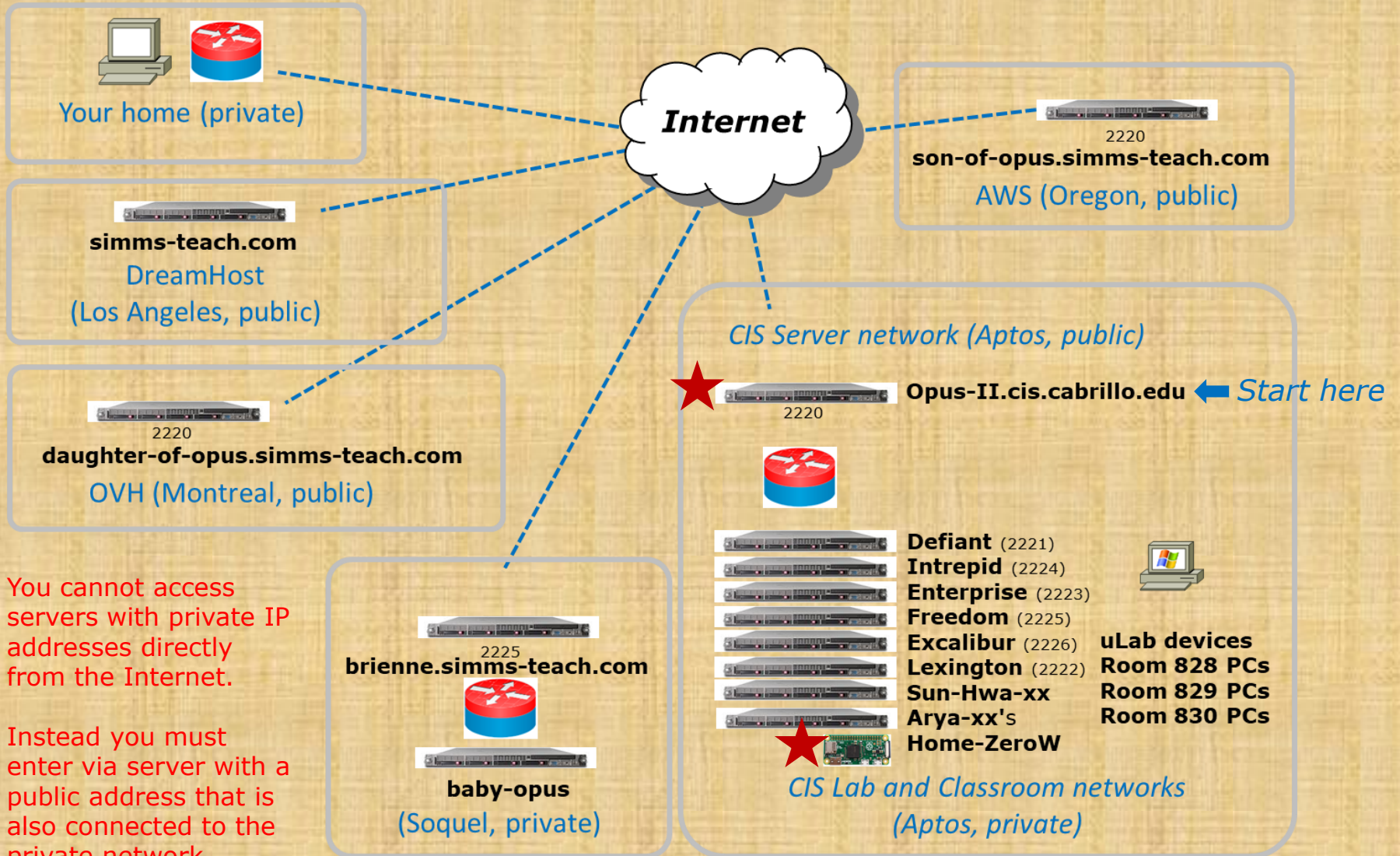
Metal Sitemap W3C XHTML 1.0 W3C CSS Credits Earth

VM Name	VM ID
Araya	Araya 1
Araya 2	Araya 2
Araya 3	Araya 3
Araya 4	Araya 4
Araya 5	Araya 5
Araya 6	Araya 6
Araya 7	Araya 7
Araya 8	Araya 8
Araya 9	Araya 9
Araya 10	Araya 10
Araya 11	Araya 11
Araya 12	Araya 12
Araya 13	Araya 13
Araya 14	Araya 14
Araya 15	Araya 15
Araya 16	Araya 16
Araya 17	Araya 17
Araya 18	Araya 18
Araya 19	Araya 19
Araya 20	Araya 20
Araya 21	Araya 21
Araya 22	Araya 22
Araya 23	Araya 23
Araya 24	Araya 24
Araya 25	Araya 25
Araya 26	Araya 26
Araya 27	Araya 27
Araya 28	Araya 28
Araya 29	Araya 29
Araya 30	Araya 30
Araya 31	Araya 31
Araya 32	Araya 32
Araya 33	Araya 33
Araya 34	Araya 34
Araya 35	Araya 35
Araya 36	Araya 36
Araya 37	Araya 37
Araya 38	Araya 38
Araya 39	Araya 39
Araya 40	Araya 40
Araya 41	Araya 41
Araya 42	Araya 42
Araya 43	Araya 43
Araya 44	Araya 44
Araya 45	Araya 45
Araya 46	Araya 46
Araya 47	Araya 47
Araya 48	Araya 48
Araya 49	Araya 49
Araya 50	Araya 50

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

Class Activity

Follow me if you can!

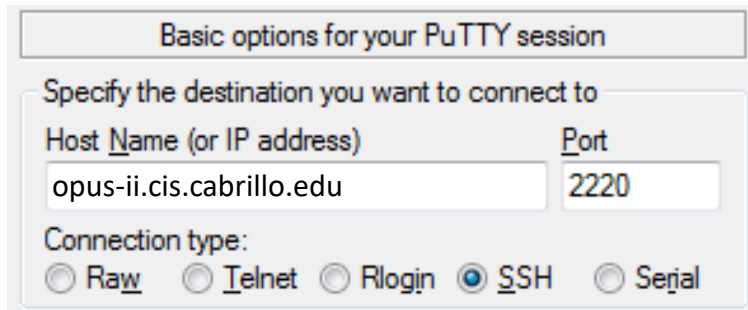


You cannot access servers with private IP addresses directly from the Internet.

Instead you must enter via server with a public address that is also connected to the private network.

FYI, specifying the username on ssh logins

Putty from Windows PCs



Basic options for your PuTTY session

Specify the destination you want to connect to

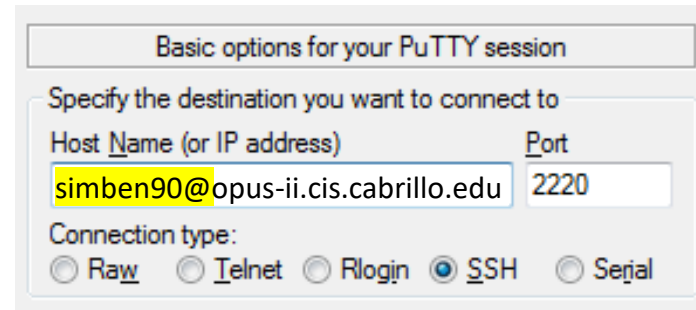
Host Name (or IP address) Port

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@opus-ii.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

Connection type:

☐ Raw ☐ Telnet ☐ Rlogin ☒ SSH ☐ Serial

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

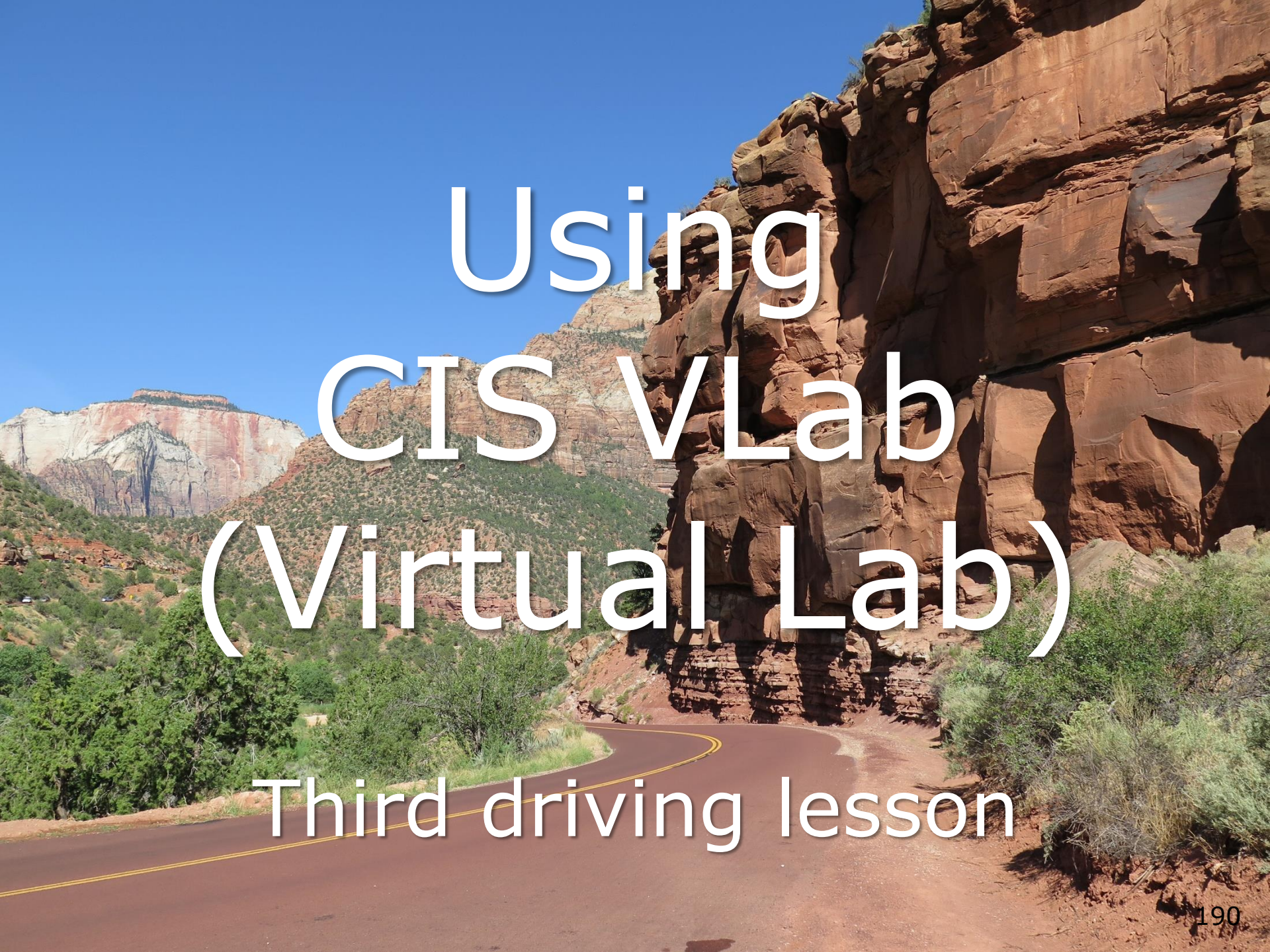
```
Using username "simben90".
simben90@opus-ii.cis.cabrillo.edu's password:
```

ssh command from Mac or UNIX/Linux systems

```
ssh -p 2220 simben90@opus-ii.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@opus-ii.cis.cabrillo.edu
simben90@opus-ii.cis.cabrillo.edu's password:
```



Using CIS VLab (Virtual Lab)

Third driving lesson

Command Line vs Graphical Desktop

Should I use SSH or VLab?

SSH when:

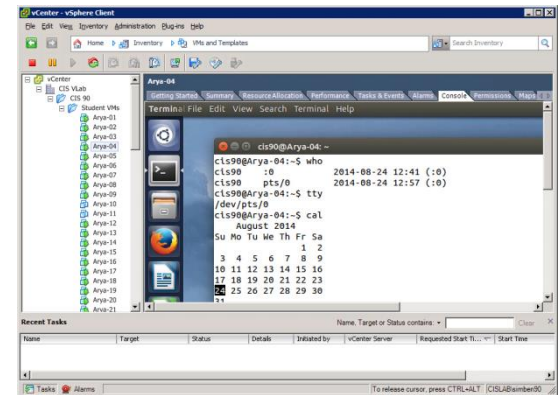
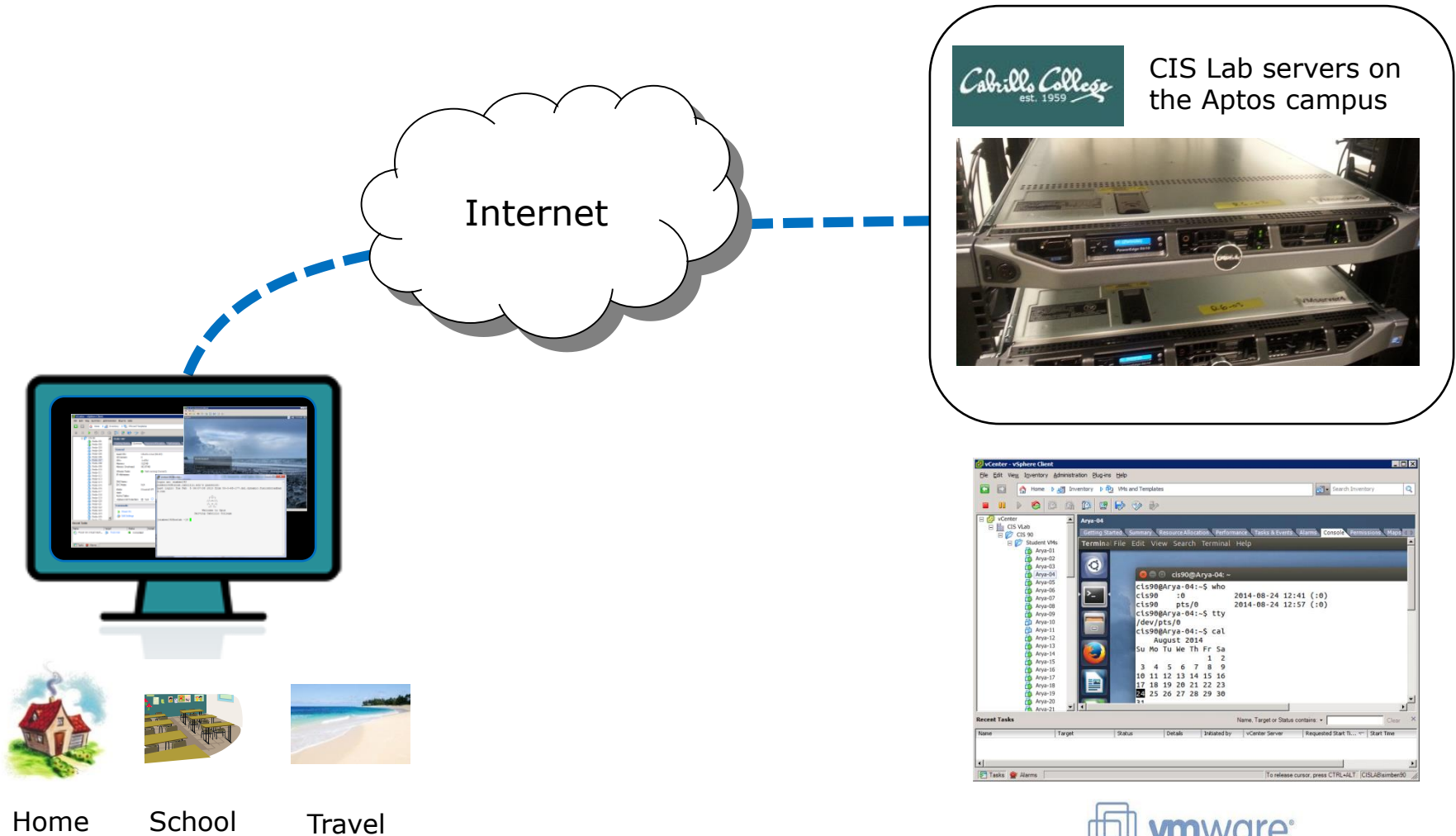
- You just need a command line.
- Have a low or high speed network connection.

VLab when:

- You need to use a graphical desktop.
- You need to use virtual terminals (the very basic black consoles).
- Higher speed network connection is needed for the graphics.

VLab = using the VMware vSphere Web Client over the Internet to access course VMs.

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 90
[Previous Terms](#)

1 day till term starts!

Cabrillo College
[Web Advisor](#)

VLab (web)
NETLAB+ VE
[Annoying Issue List](#)

CIS 90 VLab VM Assignments

[RIP Dennis Ritchie](#)

Opus Status: UP

Rich Simms

Contact

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

My Spring 2018 Cabrillo Classes

- CIS 90 - Introduction to UNIX/Linux

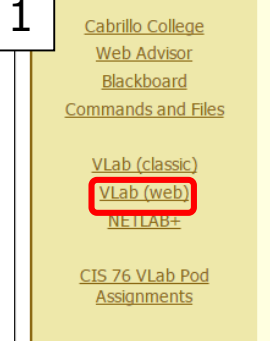
Metal Sitemap W3C XHTML 1.0 W3C CSS Credits Earth

VM Name	VM ID
Araya	Araya 1
Araya 2	Araya 2
Araya 3	Araya 3
Araya 4	Araya 4
Araya 5	Araya 5
Araya 6	Araya 6
Araya 7	Araya 7
Araya 8	Araya 8
Araya 9	Araya 9
Araya 10	Araya 10
Araya 11	Araya 11
Araya 12	Araya 12
Araya 13	Araya 13
Araya 14	Araya 14
Araya 15	Araya 15
Araya 16	Araya 16
Araya 17	Araya 17
Araya 18	Araya 18
Araya 19	Araya 19
Araya 20	Araya 20
Araya 21	Araya 21
Araya 22	Araya 22
Araya 23	Araya 23
Araya 24	Araya 24
Araya 25	Araya 25
Araya 26	Araya 26
Araya 27	Araya 27
Araya 28	Araya 28
Araya 29	Araya 29
Araya 30	Araya 30
Araya 31	Araya 31
Araya 32	Araya 32
Araya 33	Araya 33
Araya 34	Araya 34
Araya 35	Araya 35
Araya 36	Araya 36
Araya 37	Araya 37
Araya 38	Araya 38
Araya 39	Araya 39
Araya 40	Araya 40
Araya 41	Araya 41
Araya 42	Araya 42
Araya 43	Araya 43
Araya 44	Araya 44
Araya 45	Araya 45
Araya 46	Araya 46
Araya 47	Araya 47
Araya 48	Araya 48
Araya 49	Araya 49
Araya 50	Araya 50

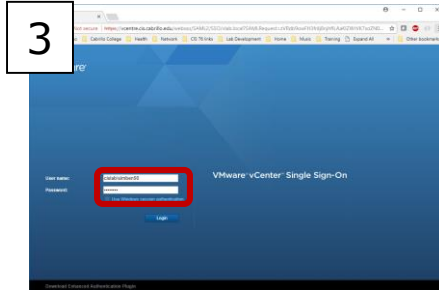
To see which Araya VM is yours use the link on the class website

Accessing CIS VLab via vSphere Web (HTML5) Client Using Chrome Browser on PC or Pac

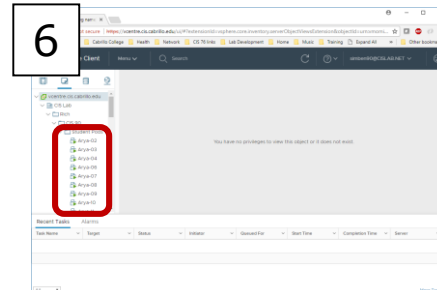
<http://simms-teach.com/>



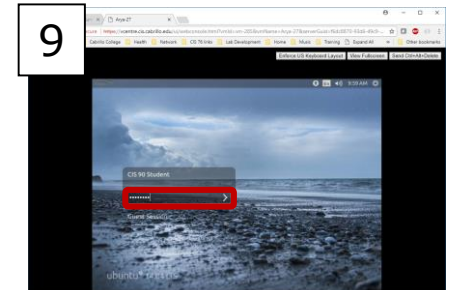
Click VLab (web) on left pane of website



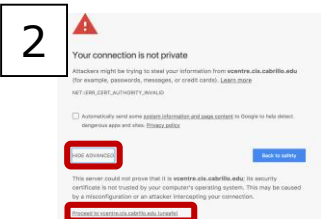
Login, username must start with cislabs\



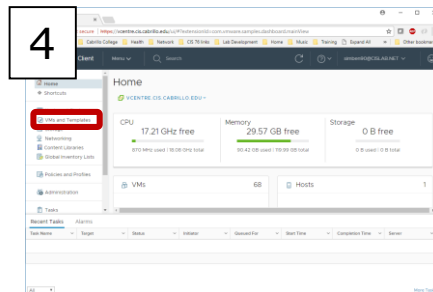
Scroll and select your Arya VM in the Student Pods folder



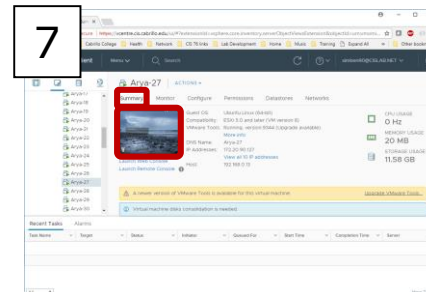
Enter password for "CIS 90 Student" (cis90 user)



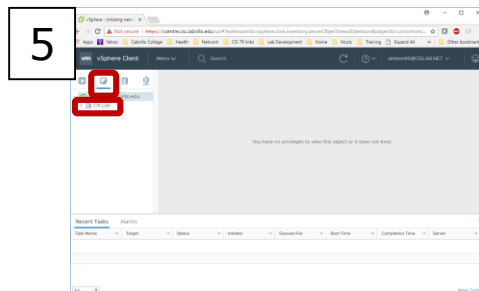
Select "Advanced" then "Proceed to vcentre ..."



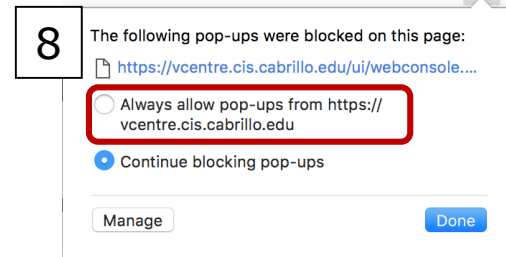
Select VMs and Templates



Under Summary tab, Click on the mini-console



Expand tree by clicking each ">" till you see Student Pods



Always allow pop-ups from vcentre

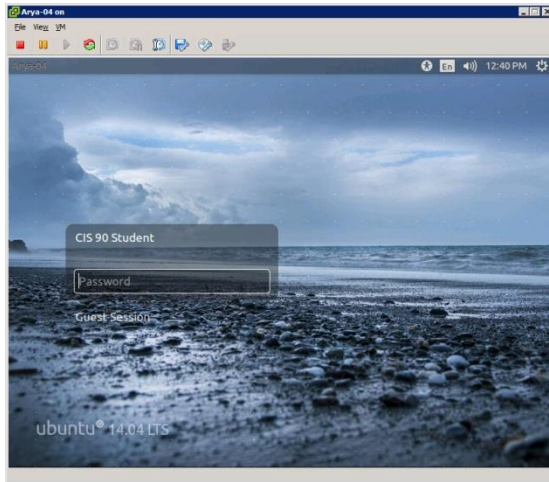
Accessing VLab Activity

Follow the instructor to open a graphical user desktop on your Arya-xx VM

- ☐ Browse to <http://simms-teach.com>
- ☐ Click VLab (web) link
- ☐ Accept warning
- ☐ Login with VLab credentials*
- ☐ Select VMs and Templates view
- ☐ Expand navigation tree
- ☐ Find your Arya VM
- ☐ Click the mini-console for your Arya-VM
- ☐ Allow browser pop-ups

*See the CIS 90 announcement in Canvas from the instructor for VLab login credentials

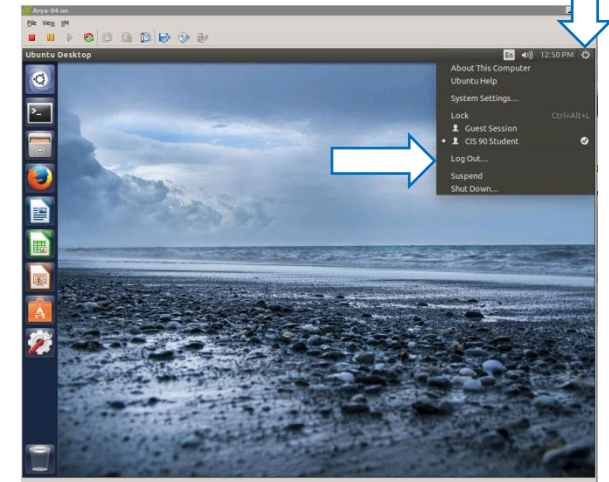
1) Log in as
CIS 90 Student



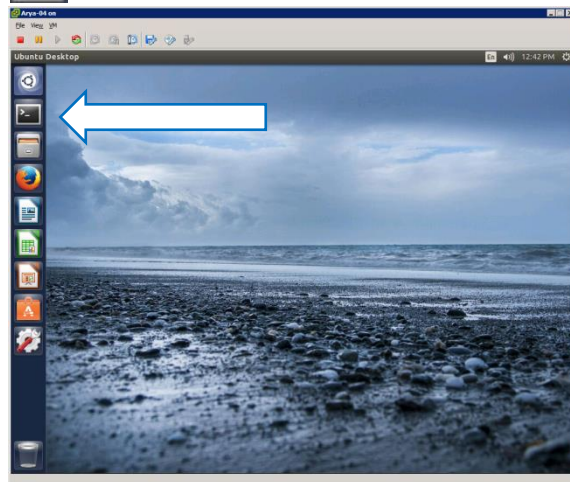
The Arya VM



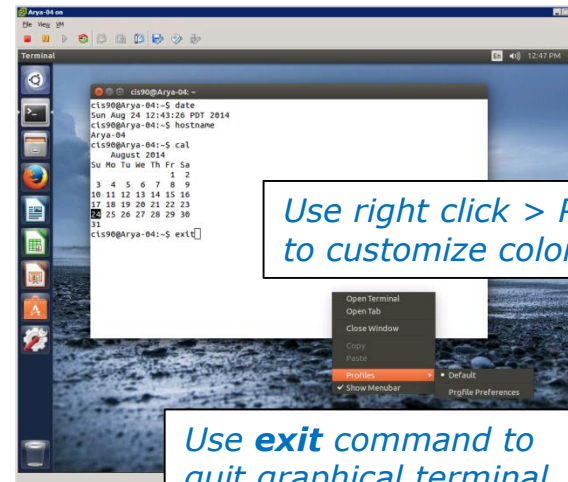
4) When finished
Gear icon > Log Out...



2) To get a graphical terminal
Terminal icon



3) Enter commands in the
graphical terminal



Using VLab VM 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 announcement in Canvas from the instructor for Arya login credentials

A photograph of a suburban street scene. The road is paved and has some cracks. On the left side, there are houses, including a light blue one and a brown one. On the right side, there is a white house with a red car parked in front of it. There are many bare trees along the street, and a utility pole is visible on the right. The sky is blue with some light clouds.

Virtual Terminals (consoles)

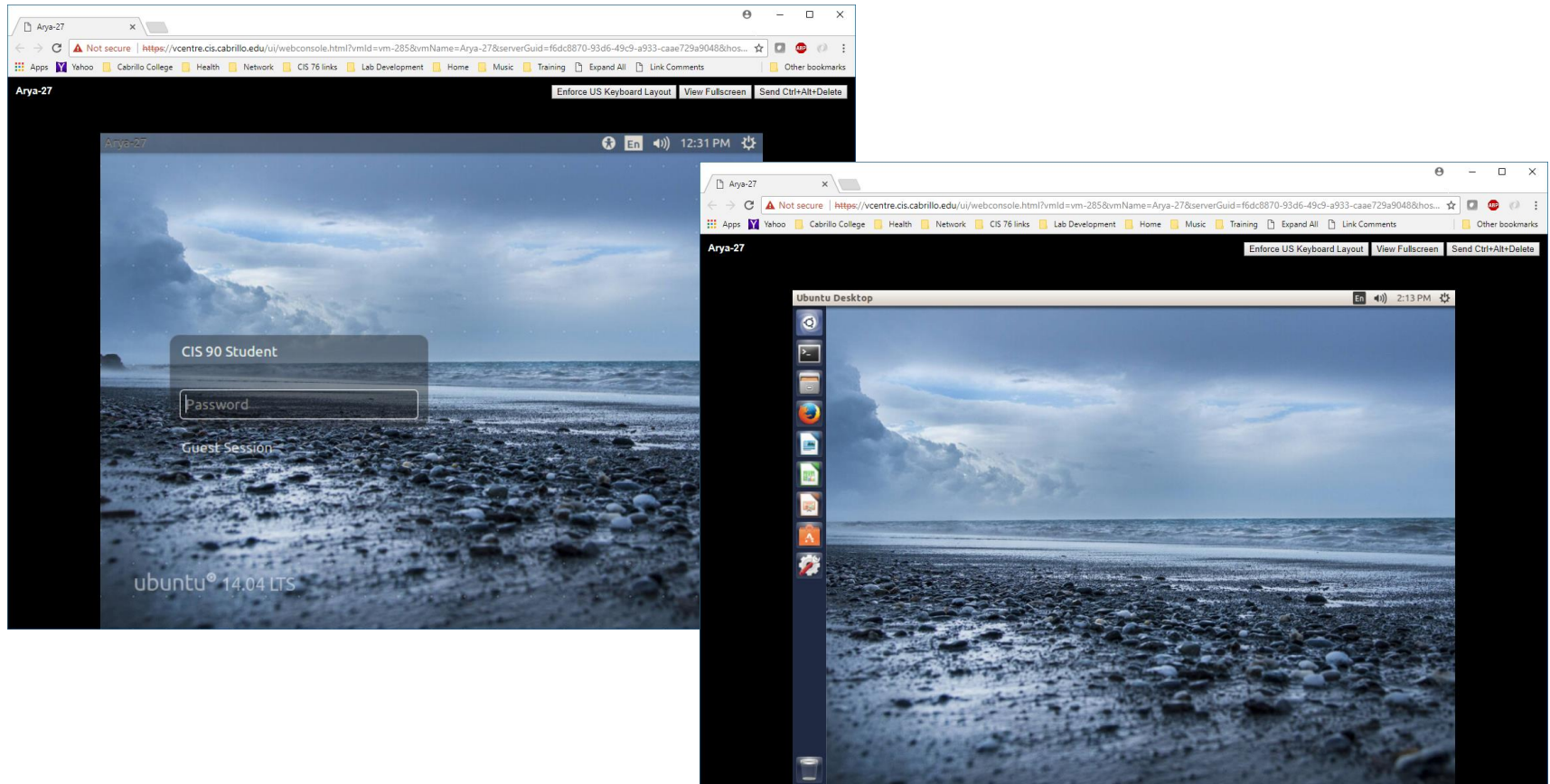
Fourth driving lesson

Remote Command Line Access (via terminal emulator with SSH)

```
cis90@Arya-27: ~  
/home/cis90/simben $ ssh cis90@arya-27  
cis90@arya-27's password:  
Welcome to Ubuntu 14.04.2 LTS (GNU/Linux 3.13.0-53-generic x86_64)  
  
 * Documentation:  https://help.ubuntu.com/  
  
526 packages can be updated.  
340 updates are security updates.  
  
      /\_/\n     /--W\ \n    /  _  \ \n   /  _  \ \n  /  _  \ \n /  _  \ \n/_/  _  \_\n\nWinter is coming
```

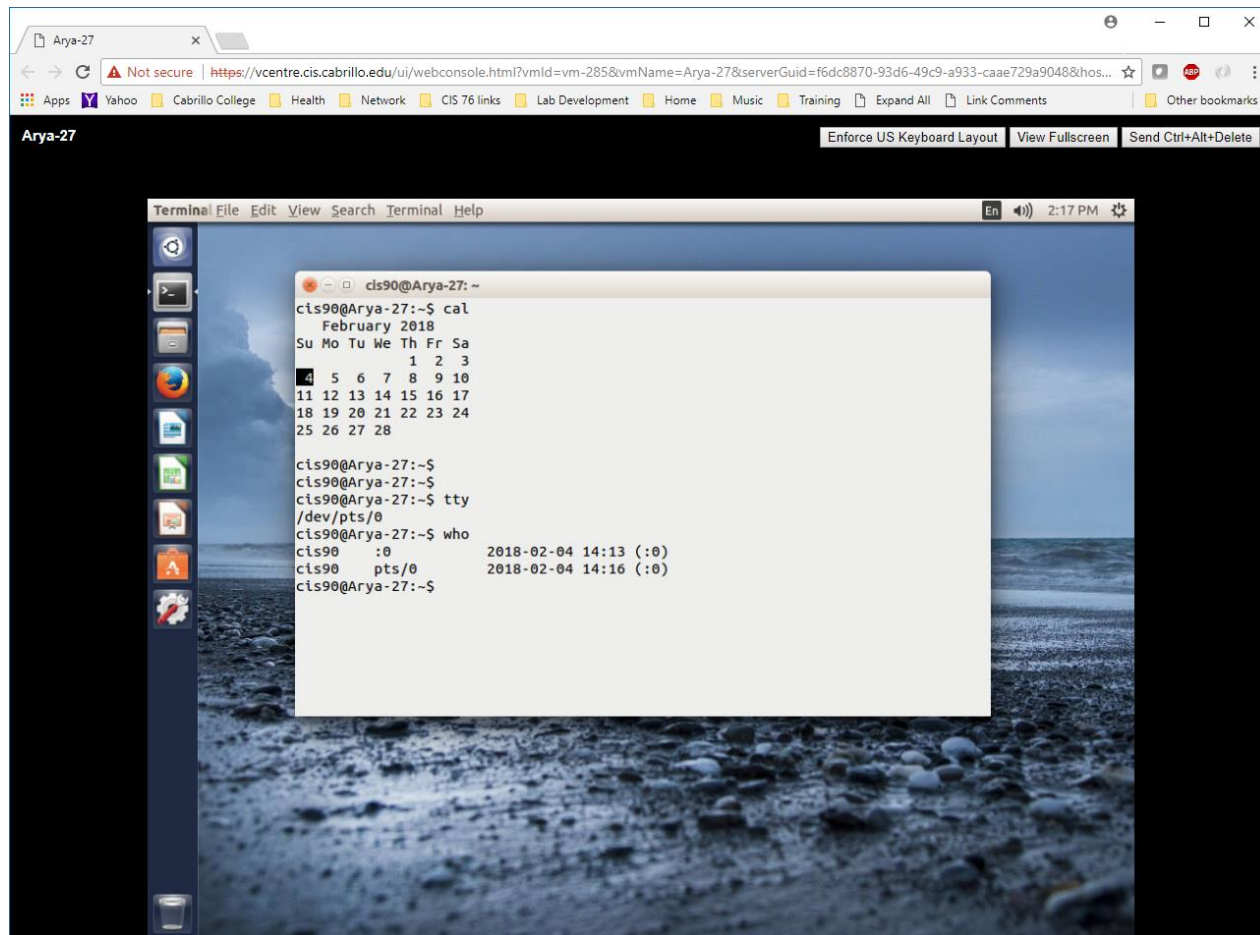
```
Last login: Sun Feb  4 12:12:26 2018 from opus-ii.cis.cabrillo.edu  
cis90@Arya-27:~$  
cis90@Arya-27:~$ cal  
February 2018  
Su Mo Tu We Th Fr Sa  
          1  2  3  
4  5  6  7  8  9 10  
11 12 13 14 15 16 17  
18 19 20 21 22 23 24  
25 26 27 28  
  
cis90@Arya-27:~$  
cis90@Arya-27:~$ tty  
/dev/pts/0  
cis90@Arya-27:~$
```

Local Graphical Desktop Access (via connected monitor or virtualization product console)



Using VLab we can login and use a graphical desktop on our Arya

Local Graphical Command Line Access (via connected monitor or virtualization product console)



Running the terminal app on Arya gives us a graphical terminal

Local Command Line access
(via connected monitor or virtualization tool)

```

Ubuntu 14.04.2 LTS Arya-27 tty2

Arya-27 login: cis90
Password:
Last login: Sun Feb  4 14:08:29 PST 2018 on tty2
Welcome to Ubuntu 14.04.2 LTS (GNU/Linux 3.13.0-53-generic x86_64)

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

526 packages can be updated.
340 updates are security updates.


      /\_/\
     /__  \
    (oo)\_____)
    (__/       )\/\
       ||----w |
       ||     ||

Winter is coming

cis90@Arya-27:~$ tty
/dev/tty2
cis90@Arya-27:~$ who
cis90          tty2            2018-02-04 14:19
cis90          :0              2018-02-04 14:13 (:0)
cis90          pts/0           2018-02-04 14:16 (:0)
cis90@Arya-27:~$ _
```

*We can also use one of several very basic TTY virtual terminals
(no mouse, no scrolling, no fonts, etc.)*

Keyboard Keys for using Virtual Terminals on VMware Linux VMs

VMware virtual terminal operations

On PC Keyboard:

While holding down the **Ctrl**--**Alt** keys, tap **spacebar** then tap **f1**, **f2**, ... or **f7**.

On Mac keyboard:

Hold down **fn**, **control** and **option** keys, tap the **spacebar**, then tap **f1**, **f2**, ... or **f7**.

Pressing the  on some Windows keyboards may not be necessary

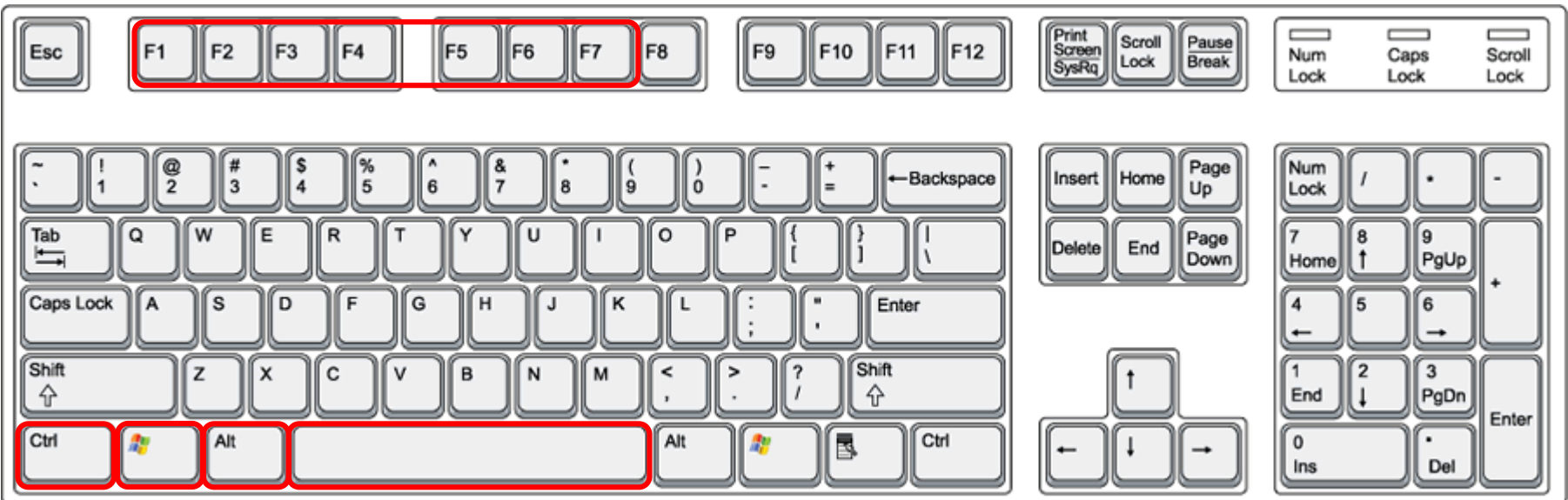
F7 is graphics mode for the Ubuntu VMs.

The Centos VMs, Like Opus-II, do not have a graphics mode components installed (run level 3 only)

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

VMware VM Operations

PC keyboard



On PC keyboard:

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

VMware VM Operations Macbook Pro keyboard



On Macbook Pro keyboard:

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

VMware VM Operations

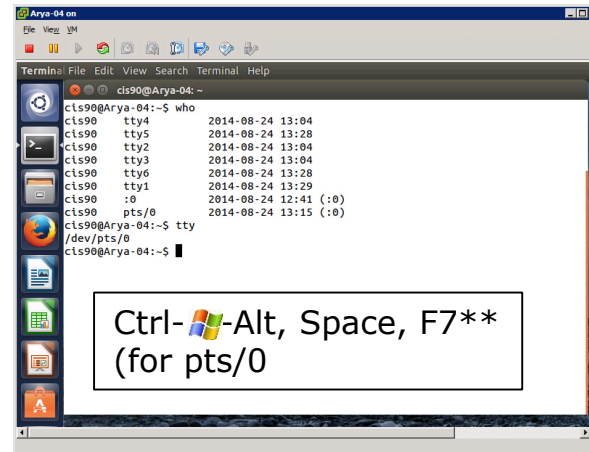
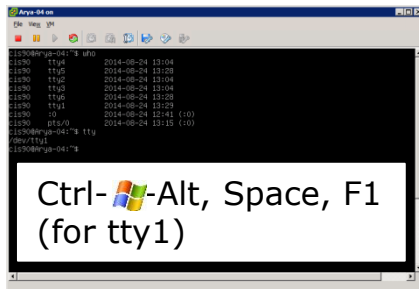
Mac keyboard



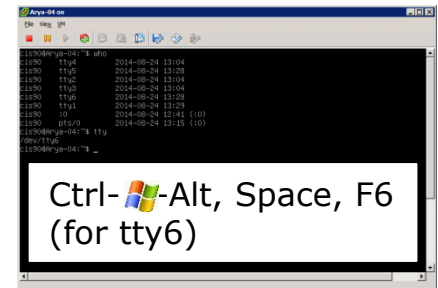
On Mac keyboard:

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

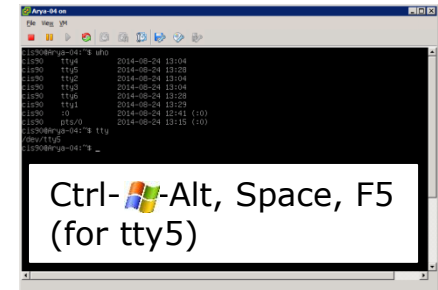
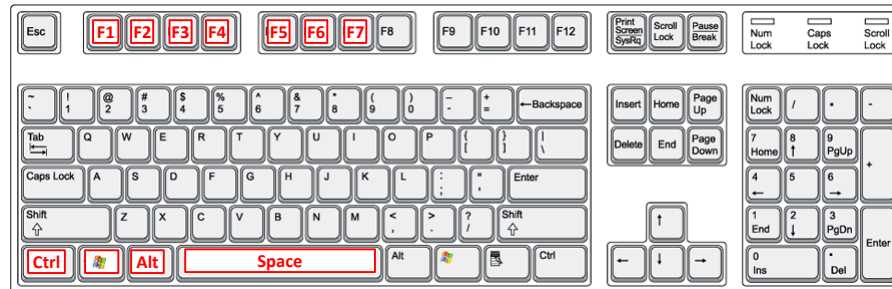
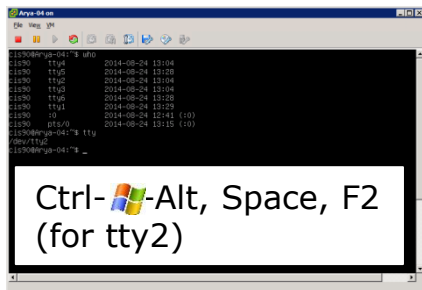
Changing Virtual TTY Terminals using VMware vSphere



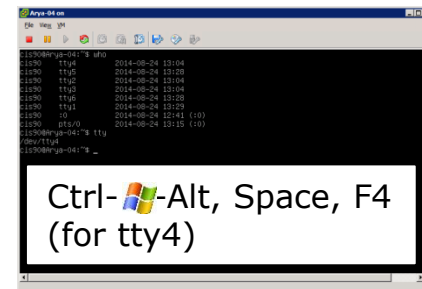
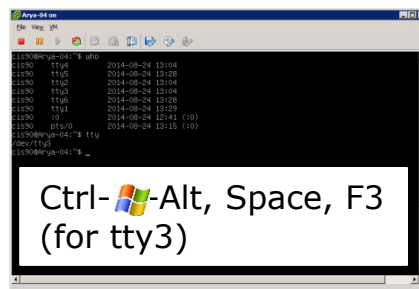
Mac users replace
ctrl--alt
with **fn-control-option**



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



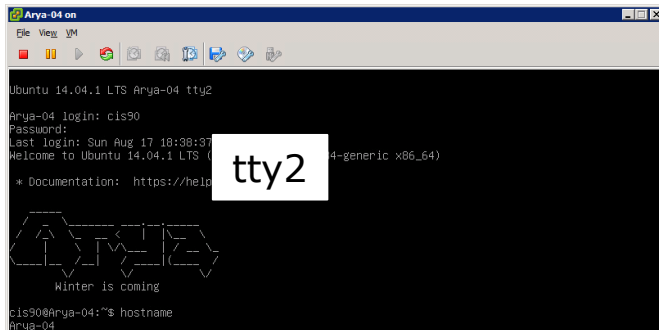
*On some PC
keyboards it is not
necessary to use
the  key



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

Virtual Terminals

- 1) While holding down **ctrl**--**alt** (PC) or **fn-control-option** (Mac) 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)



```

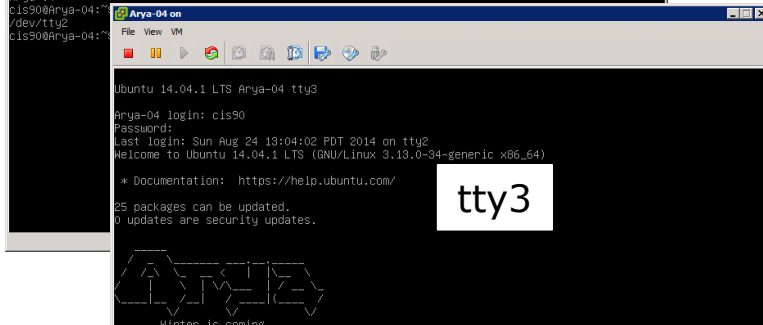
Arya-04 on
File View VM
Ubuntu 14.04.1 LTS Arya-04 tty2
Arya-04 login: cis90
Password:
Last login: Sun Aug 17 18:38:57 PDT 2014 on tty2
Welcome to Ubuntu 14.04.1 LTS (GNU/Linux 3.13.0-34-generic x86_64)

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

  ____
 /  _ \
/  / \  \
/  /  __ \
/___/  /_/_/
Winter is coming

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

```



```

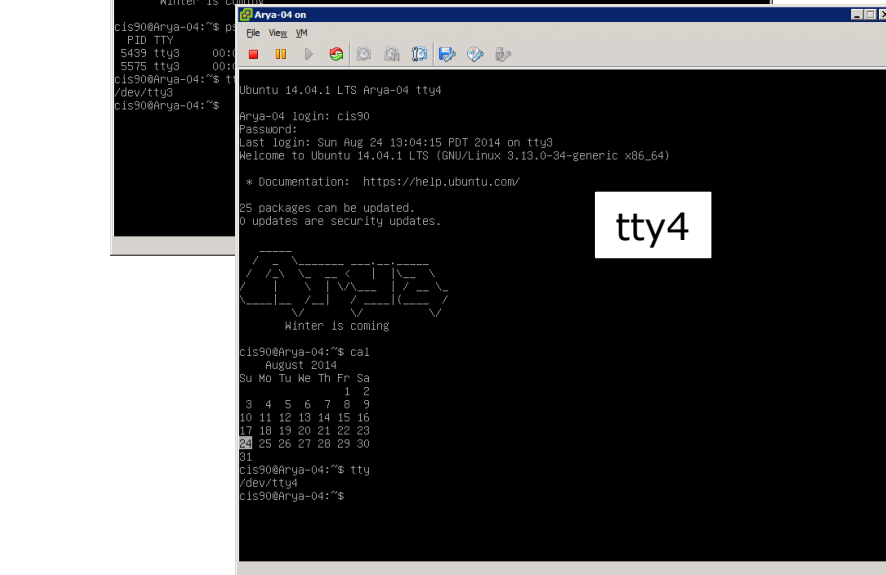
Arya-04 on
File View VM
Ubuntu 14.04.1 LTS Arya-04 tty3
Arya-04 login: cis90
Password:
Last login: Sun Aug 24 13:04:02 PDT 2014 on tty2
Welcome to Ubuntu 14.04.1 LTS (GNU/Linux 3.13.0-34-generic x86_64)

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

  ____
 /  _ \
/  / \  \
/  /  __ \
/___/  /_/_/
Winter is coming

cis90@Arya-04:~$

```



```

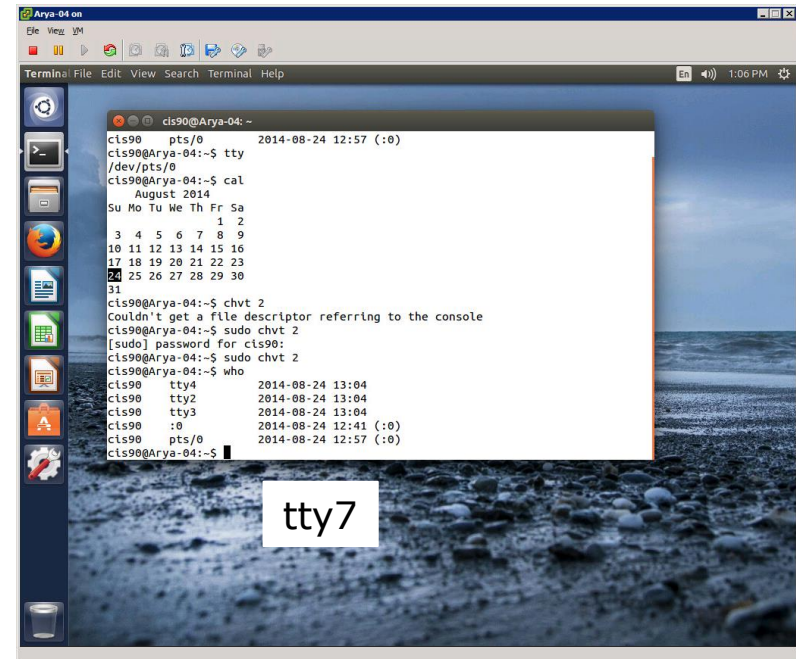
Arya-04 on
File View VM
Ubuntu 14.04.1 LTS Arya-04 tty4
Arya-04 login: cis90
Password:
Last login: Sun Aug 24 13:04:15 PDT 2014 on tty3
Welcome to Ubuntu 14.04.1 LTS (GNU/Linux 3.13.0-34-generic x86_64)

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

  ____
 /  _ \
/  / \  \
/  /  __ \
/___/  /_/_/
Winter is coming

cis90@Arya-04:~$ cal
August 2014
Su Mo Tu We Th Fr Sa
 1  2
 3  4  5  6  7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30
31
cis90@Arya-04:~$ tty
/dev/tty4
cis90@Arya-04:~$

```



```

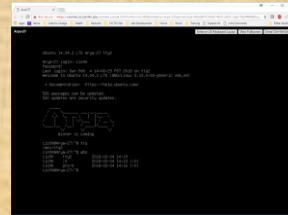
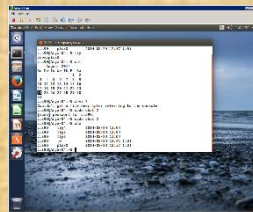
Arya-04 on
File View VM
Terminal File Edit View Search Terminal Help
cis90@Arya-04:~$
cis90 pts/0 2014-08-24 12:57 (:0)
cis90@Arya-04:~$ tty
/dev/pts/0
cis90@Arya-04:~$ cal
August 2014
Su Mo Tu We Th Fr Sa
 1  2
 3  4  5  6  7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30
31
cis90@Arya-04:~$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

```

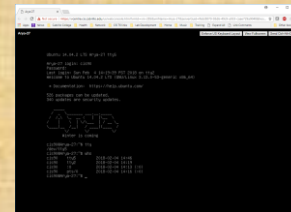
Virtual Terminal Class Activity

Mac users replace
ctrl--alt
with **fn-control-option**

ctrl--alt-Space-f2



ctrl--alt-space-f5



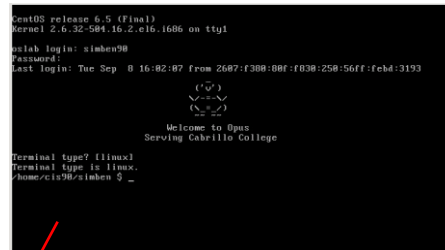
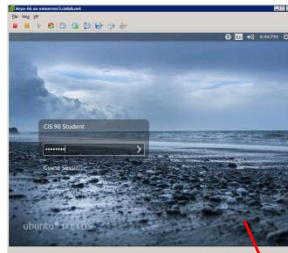
ctrl--alt-space-f7

On your Arya:

- Try changing between the graphical desktop and the TTYS
- Login as cis90 on tty2 and tty5
- Run a terminal on the graphical desktop
- Use the who command to see how many logins there are

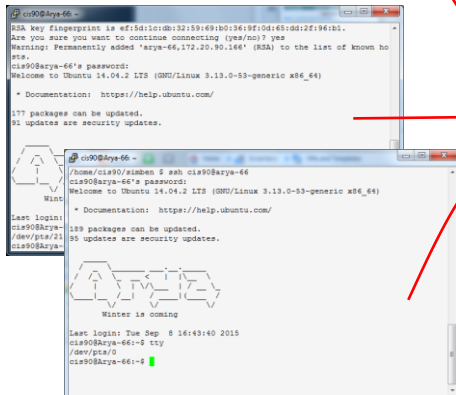
Interpreting who output

:0



tty1

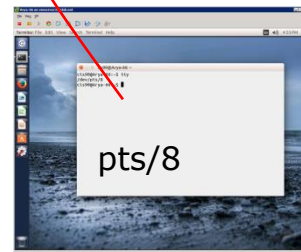
pts/21



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)

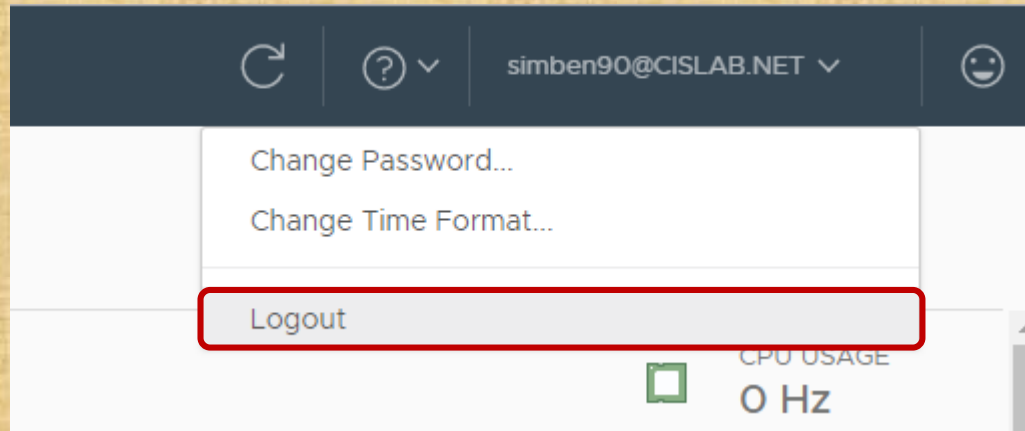


pts/8

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 out of VLab

Logging out of Arya Desktop Activity



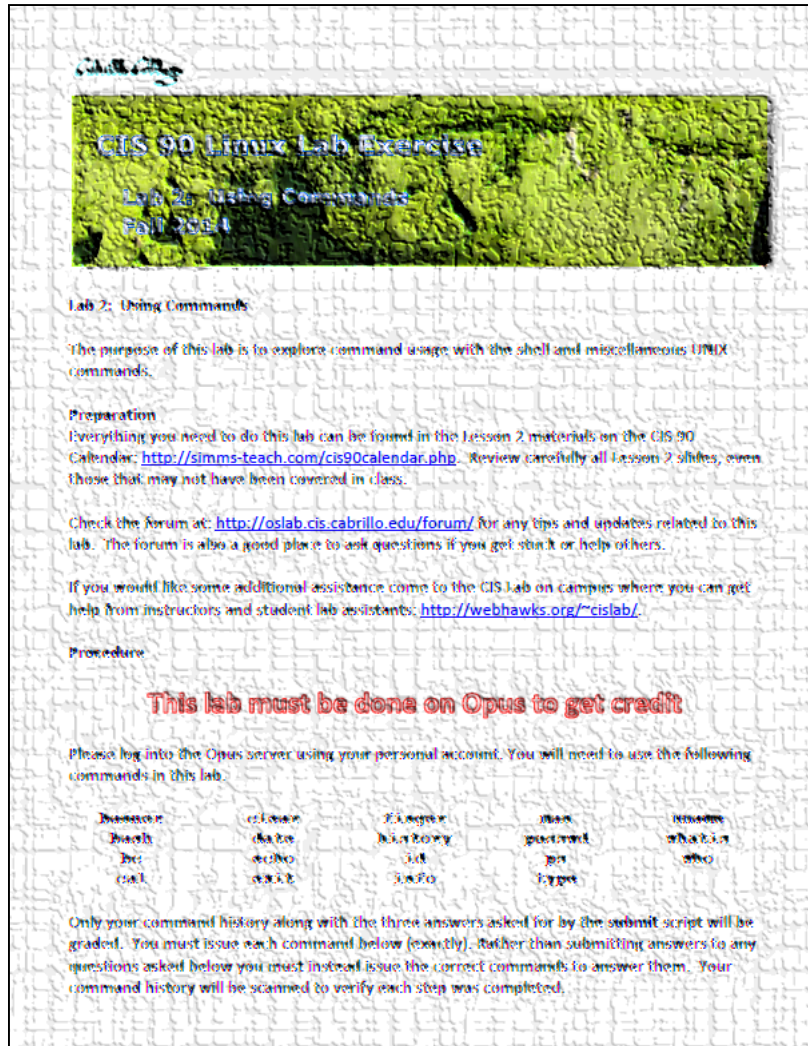
Logout of Vlab's vCenter

Your Arya VM will keep running even though you disconnect from vCenter

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	cd /usr/bin	man	rm -rf
ls	ls	ls -l	man	rm -rf
find	find	find	man	rm -rf
grep	grep	grep	man	rm -rf

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
/usr/local/bin	- custom local commands

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?

End Meeting

End Meeting

Backup

FYI

CIS 90 and Smartphones (Android)



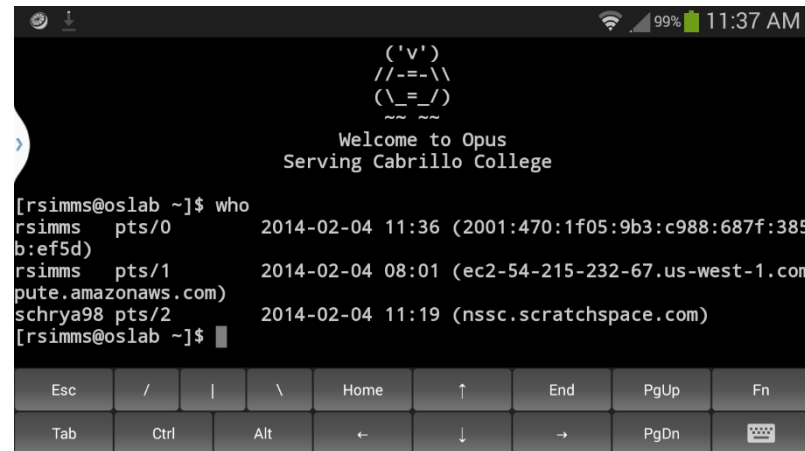
Blackboard
Collaborate App



*Join CCC Confer
virtual classroom*



JuiceSSH - SSH Client app

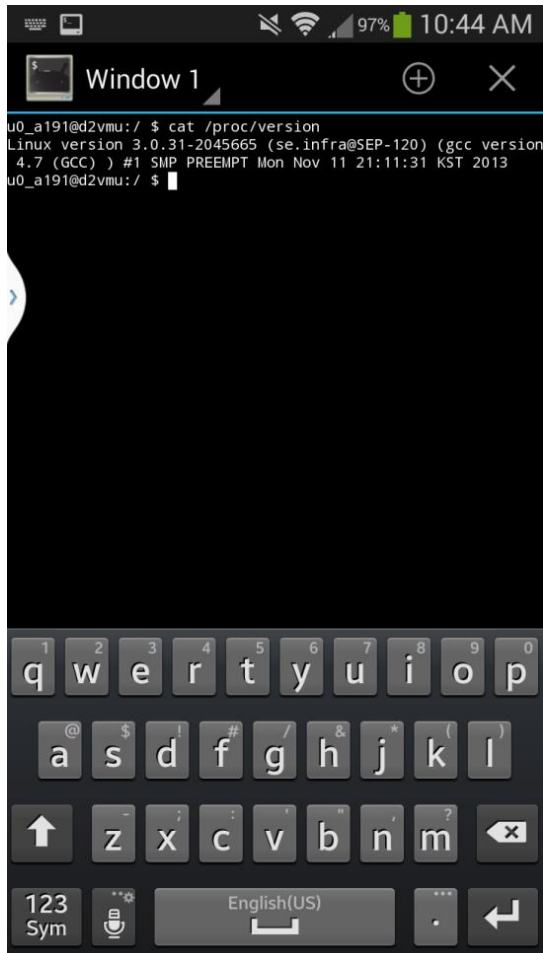


Login to to Opus

CIS 90 and Smartphones (Android)



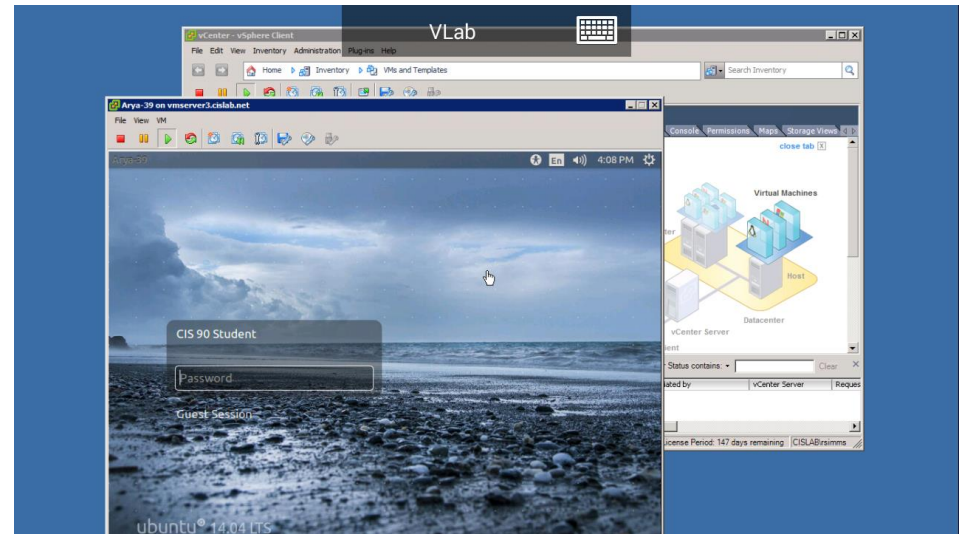
Android Terminal App



*Viewing kernel version on
smartphone*



Microsoft RDP App



Running Arya VM in VLab

Terminals

Hardware Terminals



Teletype (TTY)



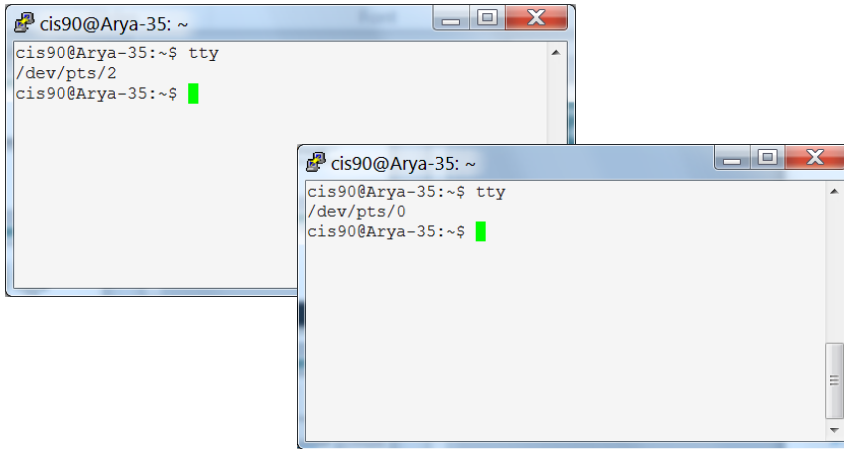
VT100



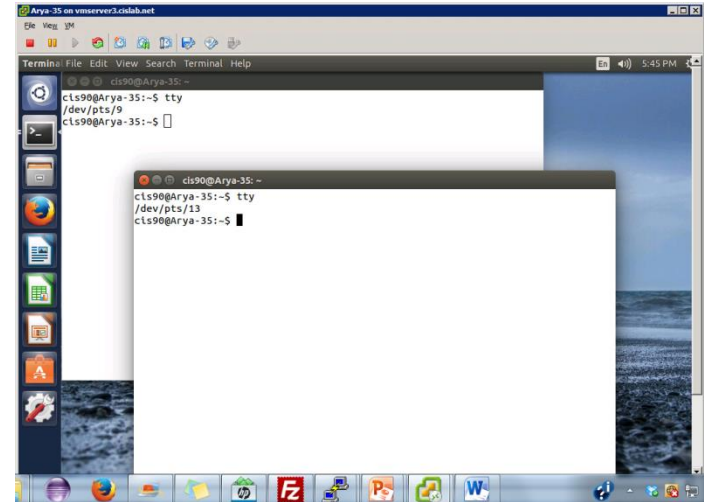
Terminals were used in the old days to interact with "minicomputers" and "mainframe" computers.

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

Software Terminals



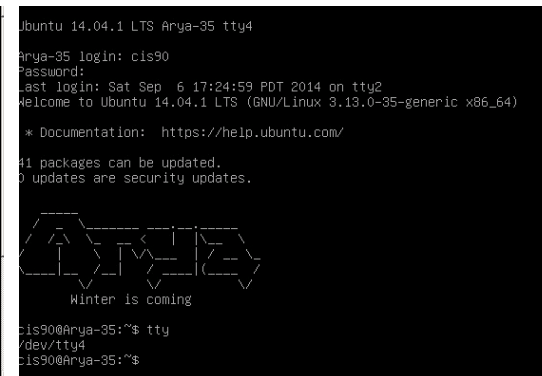
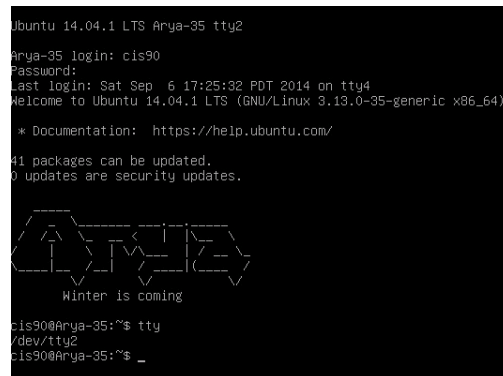
Terminal emulators like PuTTY (with scroll bars, colors, customizable backgrounds, fonts and sizes) for Windows



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

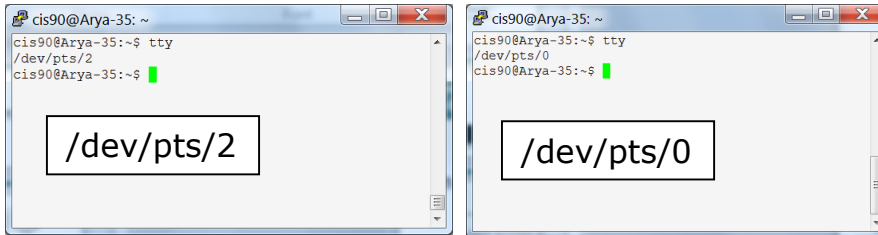
Virtual terminals (use ctrl-alt-fn)

Bare bones, no scroll bars,
also called a console



Various terminal devices on an Arya VM

Terminal emulators (e.g. Putty)



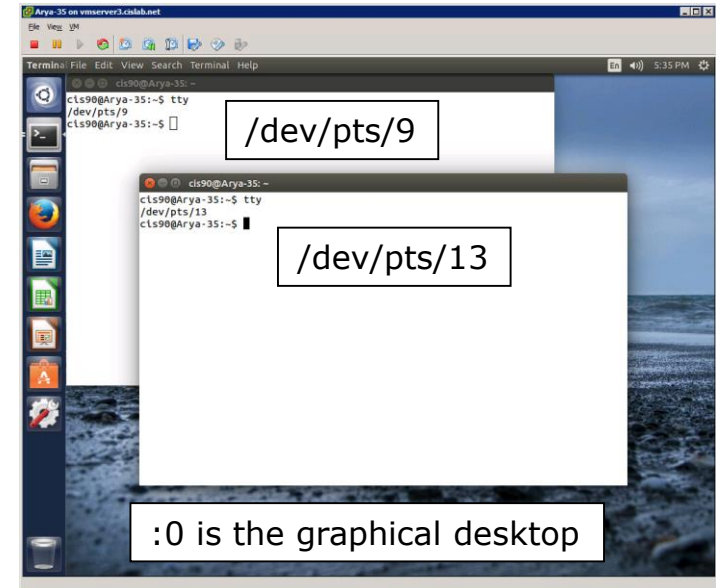
```
cis90@Arya-35:~$ who
cis90    tty4      2014-09-06 17:25
cis90    tty2      2014-09-06 17:25
cis90    pts/2      2014-09-06 17:20 (enterprise.cis.cabrillo.edu)
cis90    :0        2014-09-06 17:20 (:0)
cis90    pts/0      2014-09-06 17:21 (2601:9:6680:53b:4d09:e2b6:e7fc:d999)
cis90    pts/9      2014-09-06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

pts=pseudo terminal,

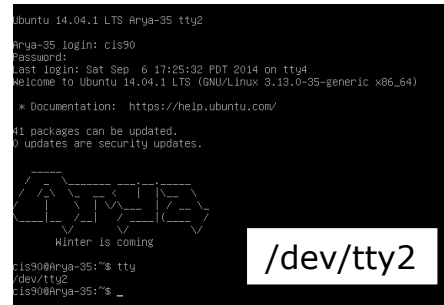
tty=teletype

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

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



Rich's Cabrillo College CIS Classes Resources

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

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

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

102 days till
term ends!

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

Links

Instructors

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

Getting Linux

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

Tools and Software

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

Standards

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

Commands

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

Clubs

- [GNU Linux Users Group](#)

Departments

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

Crib Sheets

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

Documentation

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

Animations

- [Linux network technologies](#)

Rich's Howtos

Putty

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

VirtualBox

- [Bringing the Eko VM home](#)

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

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

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



Linux Howtos

Configuring the appearance of PuTTY

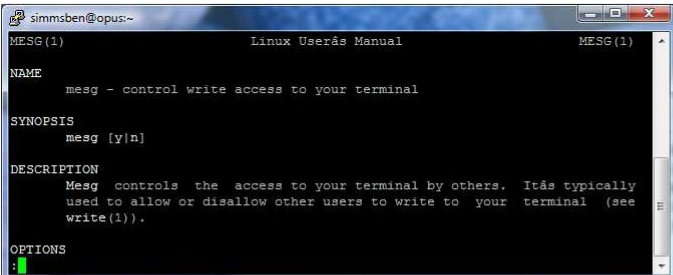
Fall, 2008

Software used

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

Step 1 - Run PuTTY and login

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



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

SYNOPSIS
    msg [y|n]

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

OPTIONS
```

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

Step 2 - Get to Reconfiguration window



```
simmsben@opus:~$
meeting (5).jnlp
meeting (4).jnlp
meeting (3).jnlp
Show all downloads...
```



Lesson 1 Review

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

```
login as: simben90
simben90@opus-ii.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@opus-ii.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@opus-ii.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@opus-ii.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  
opus-ii.cabrillo.edu  
/home/cis90/simben $
```

Answer: opus-ii.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.