Rich's lesson module checklist

Last updated 11/07/2018

- ❑ Zoom recording named and published for previous lesson

- ❑ Slides and lab posted
- ❑ Print out agenda slide and annotate page numbers

- ❑ Flash cards
- ❑ 1st minute quiz
- ❑ Calendar page updated

- ❑ Lab 9 tested and uploaded
- ❑ Test 2 stats run
- ❑ Test and schedule langs file email for Lab 9 ready (at end of class)
- ❑ Schedule lock/unlock turnin directory (scripts/schedule-submit-locks)
- ❑ Apache configured for student websites
  - ❑ /etc/httpd/conf.d/userdir.conf
    - ❑ UserDir directive
  - ❑ systemctl restart httpd
  - ❑ setsebool -P httpd_enable_homedirs true
  - ❑ chcon -R -t httpd_sys_content_t cis90_html
- ❑ Swap all egg & treat slides in shell six steps

- ❑ Backup slides, CCC info, handouts on flash drive
- ❑ Spare 9v battery for mic
- ❑ Key card for classroom 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

Shell commands

Permissions

Secure logins

Processes

Navigate file tree

Scheduling tasks

Files and directories

Mail

vi editor

Environment variables

Shell scripting

Filters

Pipes

CIS 90 Introduction to UNIX/Linux

The Command Line

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

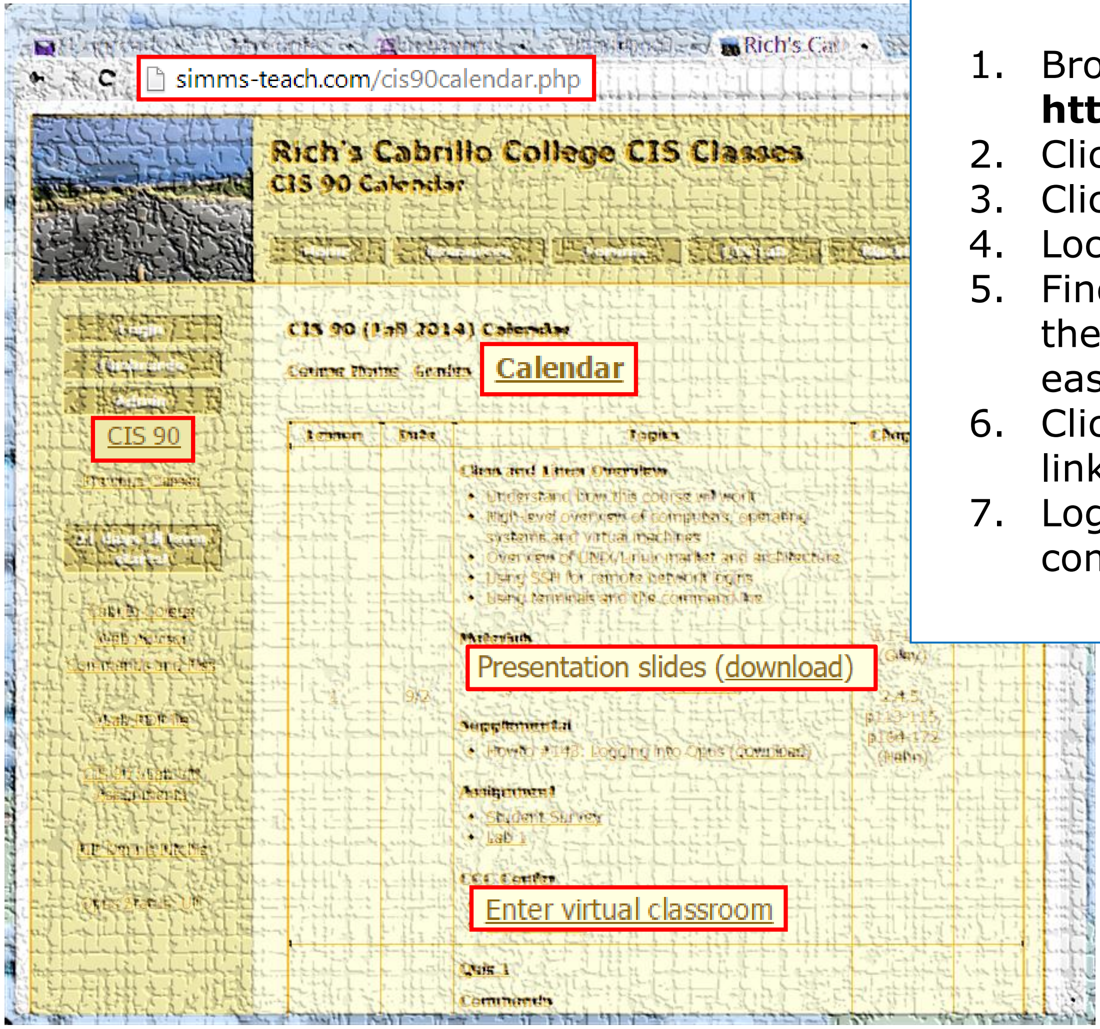
Jim Griffin
- Created this Linux course
- Created Opus and the CIS VLab
- Jim’s site: https://web.archive.org/web/20140209023942/http://cabrillo.edu/~jgriffin/

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.

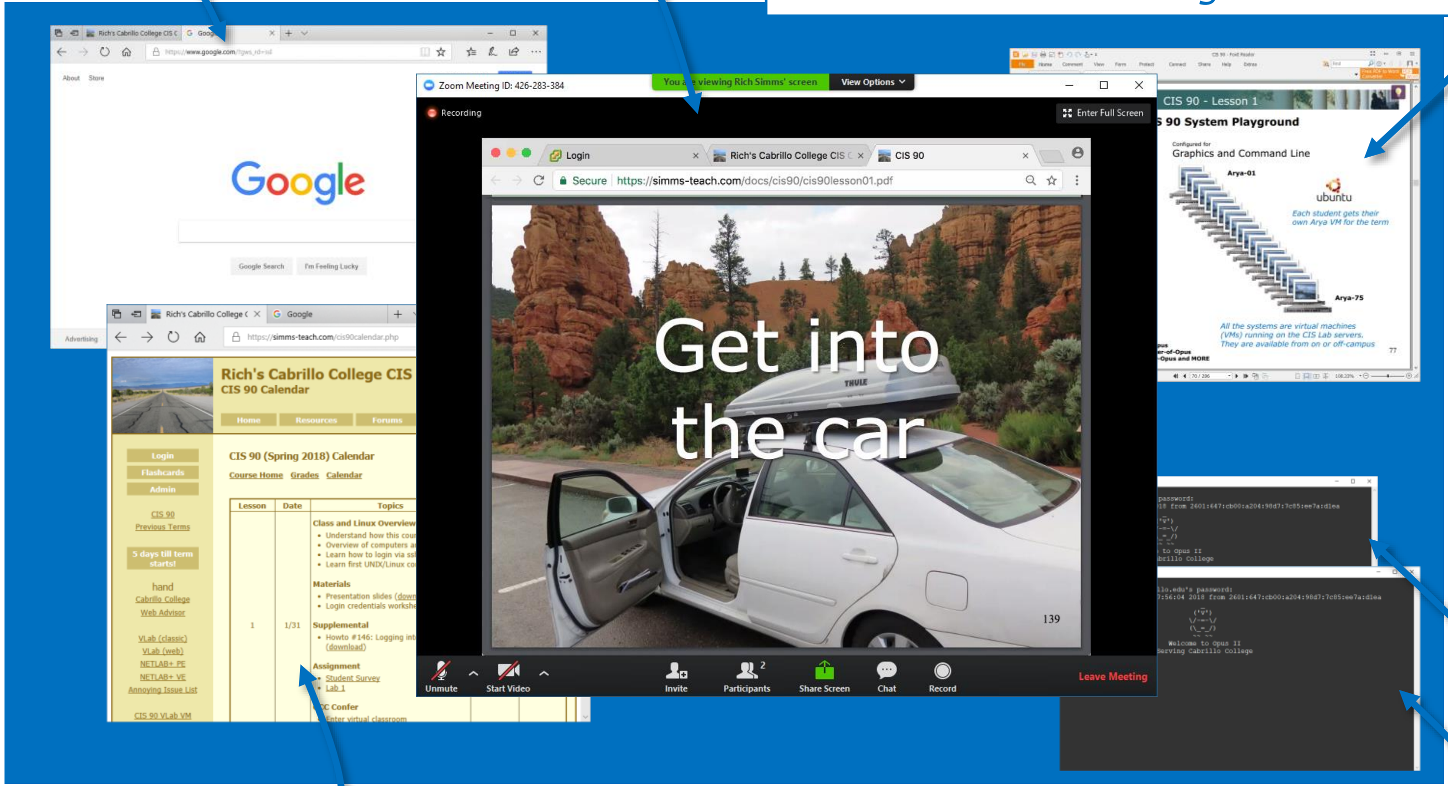
Student checklist - Before class starts


simms-teach.com/cis90calendar.php
Rich's Cabrillo College CIS Classes
CIS 90 Calendar
CIS 90 (Fall 2014) Calendar
Calendar
CIS 90
Presentation slides (download)
Enter virtual classroom

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.

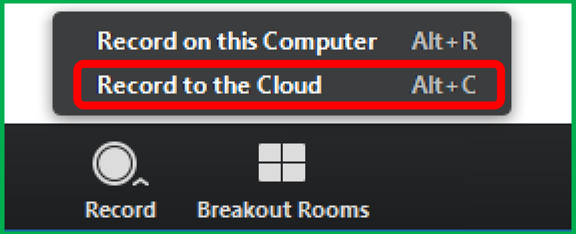
Student checklist - Before class starts

- ❑ *Google*
- ❑ *ConferZoom*
- ❑ *Downloaded PDF of Lesson Slides. I like Foxit Reader so I can take notes using annotations.*



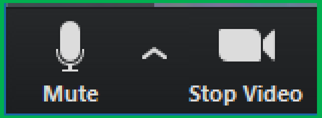
- ❑ *CIS 90 website Calendar page*
- ❑ *One or more login sessions to Opus-II*

Start



Start Recording

Audio Check



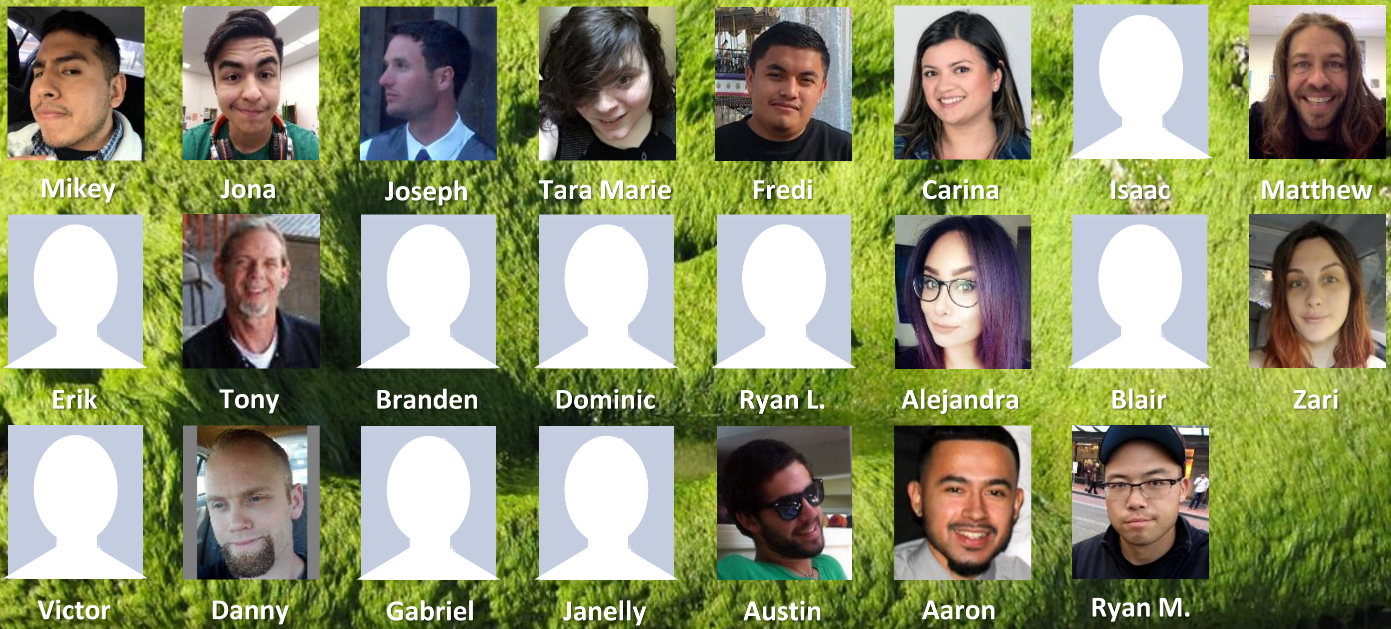
Start Recording

Audio & video Check

Cabrillo College est. 1959

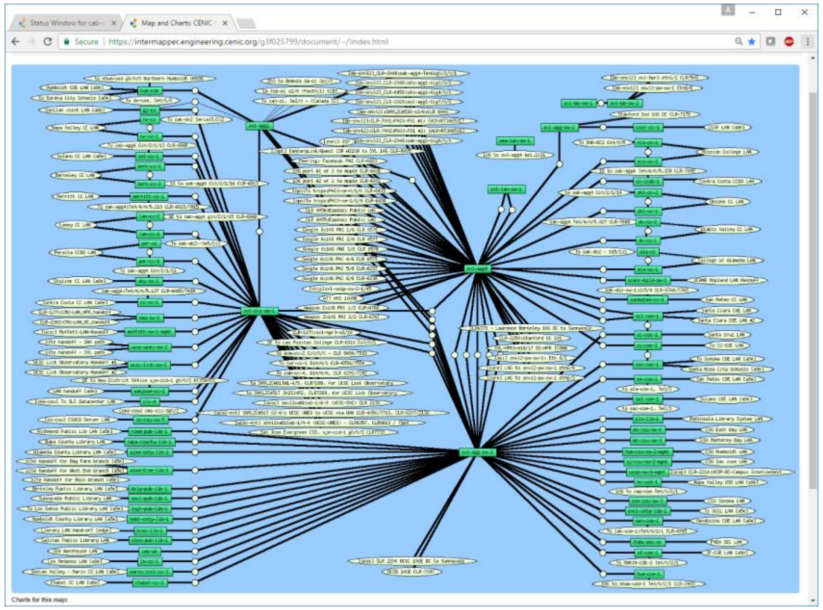
CIS 90 - Lesson 11

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



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

Network Check


Status Window for cab-
Map and Charts: CENIC
https://intermapper.engineering.cenic.org/g3f025799/document/~/!index.html
Charts for this map:

https://intermapper.engineering.cenic.org/g3f025799/document/~/!index.html

First Minute Quiz

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

Use ConferZoom White Board

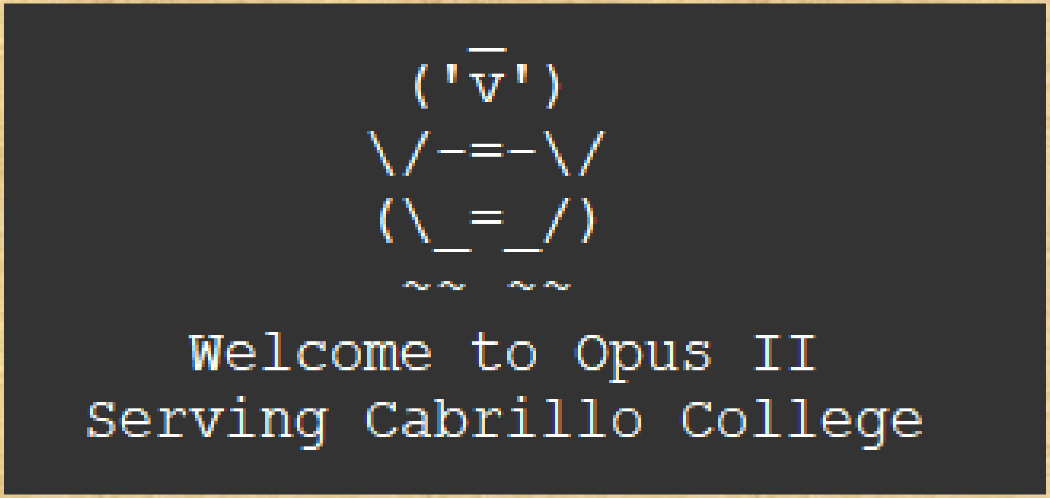
email answers to: risimms@cabrillo.edu

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

vi editor

Objectives	Agenda
• Create and modify text files	• Quiz • Questions • Test 2 Post Mortem • Housekeeping • grep workout • Shell Six Steps (review) • Signals (review) • Target Practice • Using & • Job control (review) • Load balancing & scheduling (review) • Text editors • vi 101 • vi • Tangent on spell • Assignment • Wrap up

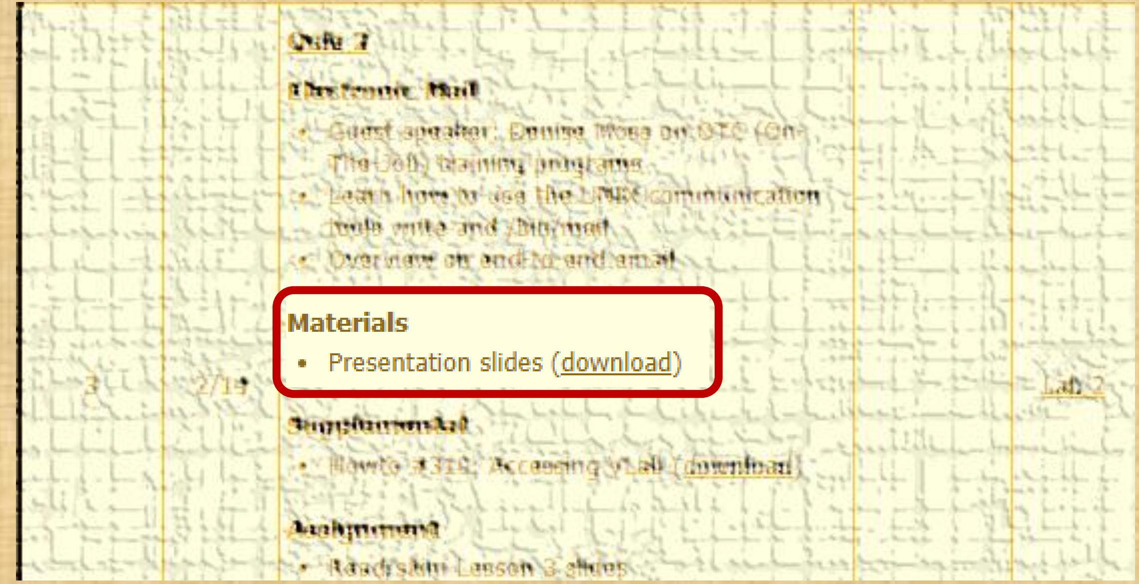
Class Activity



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

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

Class Activity

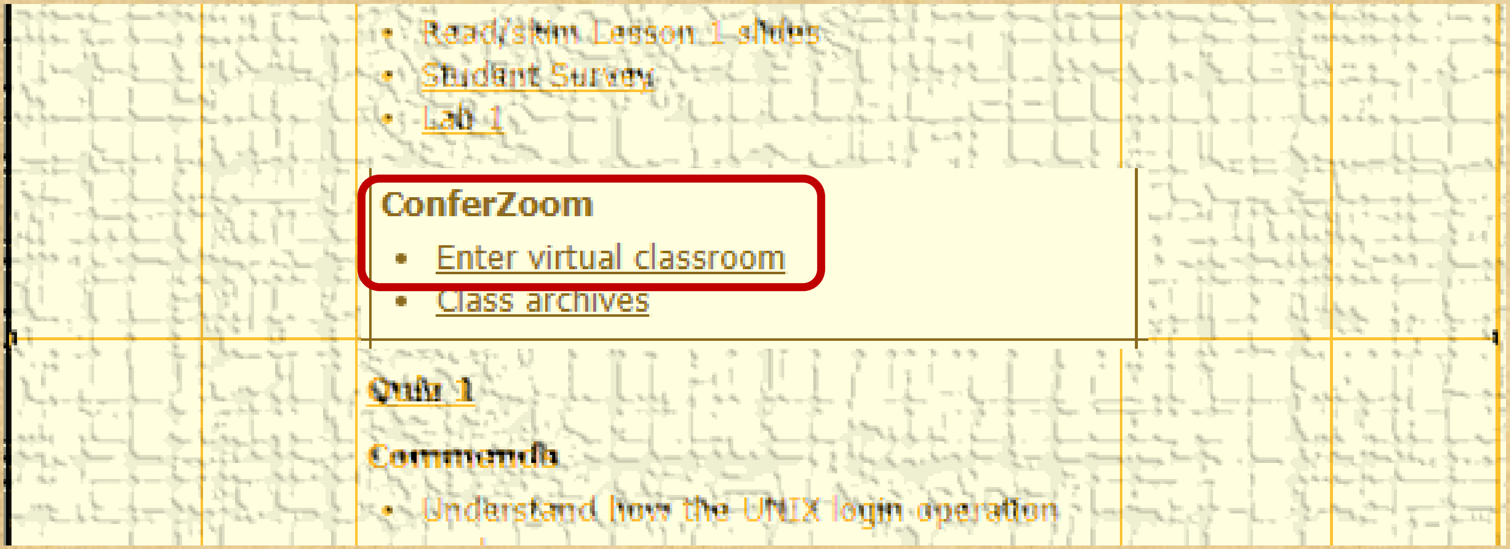


3	2/19	Quiz 7 **Electronic Mail** • Guest speaker: Denise Moss on OTC (On-The-Job) training programs • Learn how to use the UNIX communication tools write and /bin/mail • Overview on end-to-end email **Materials** • Presentation slides (download) **Supplemental** • Howto #318: Accessing VLab (download) **Assignment** • Read slides Lesson 3 slides		Lab 2

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

If you haven't already, download the lesson slides

Class Activity



- Read/skim Lesson 1 slides
- Student Survey
- Lab 1

ConferZoom

- Enter virtual classroom
- Class archives

Quiz 1

Commands

- Understand how the UNIX login operation

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

If you haven't already, join ConferZoom classroom

Questions

Questions?

Lesson material?

Labs? Tests?

How this course works?

- Graded work & tests in home directories
- Answers in /home/cis90/answers

Who questions much, shall learn much, and retain much.
- Francis Bacon

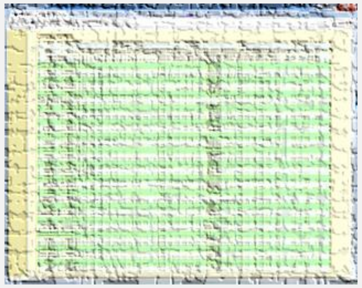
If you don't ask, you don't get.
- Mahatma Gandhi

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.*

Review your progress in the course

Check the website Grades page

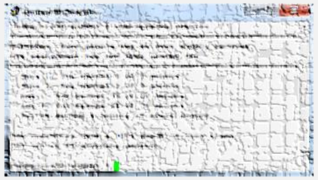
http://simms-teach.com/cis90grades.php



Or check on Opus-II

`checkgrades` *`codename`*

(where codename is your LOR codename)



Written by Jesse Warren a past CIS 90 Alumnus

- **Send me your survey to get your LOR codename.**
- **Graded labs and tests are in your home directories.**

Percentage	Total Points	Letter Grade	Pass/No Pass
90% or higher	504 or higher	A	Pass
80% to 89.9%	448 to 503	B	Pass
70% to 79.9%	392 to 447	C	Pass
60% to 69.9%	336 to 391	D	No pass
0% to 59.9%	0 to 335	F	No pass

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

Points that could have been earned:

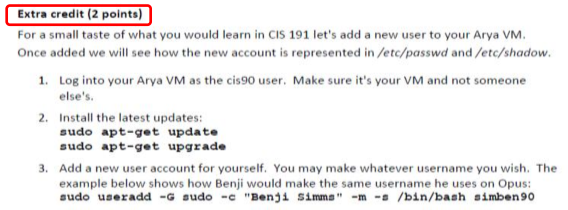
7 quizzes:	21 points
7 labs:	210 points
2 tests:	60 points
2 forum quarters:	40 points
Total:	**331 points**

Extra Credit

On the forum

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

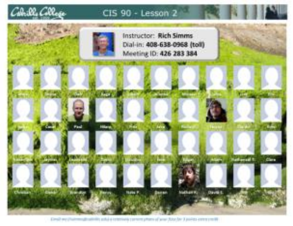
On some labs


Extra credit (2 points)

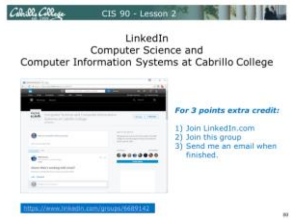
For a small taste of what you would learn in CIS 191 let's add a new user to your Arya VM. Once added we will see how the new account is represented in */etc/passwd* and */etc/shadow*.

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)


CIS 90 - Lesson 2

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


CIS 90 - Lesson 2

LinkedIn
Computer Science and
Computer Information Systems at Cabrillo College

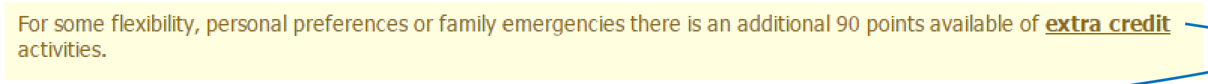
For 3 points extra credit:

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

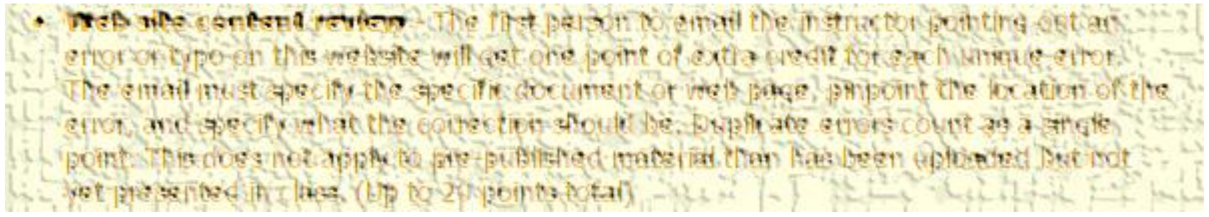
https://www.linkedin.com/groups/6689142

On the website

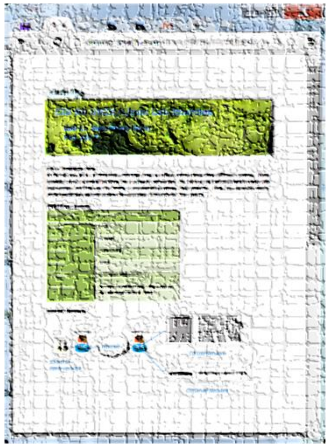
http://simms-teach.com/cis90grades.php


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


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

Lab Assignments -- Pearls of Wisdom



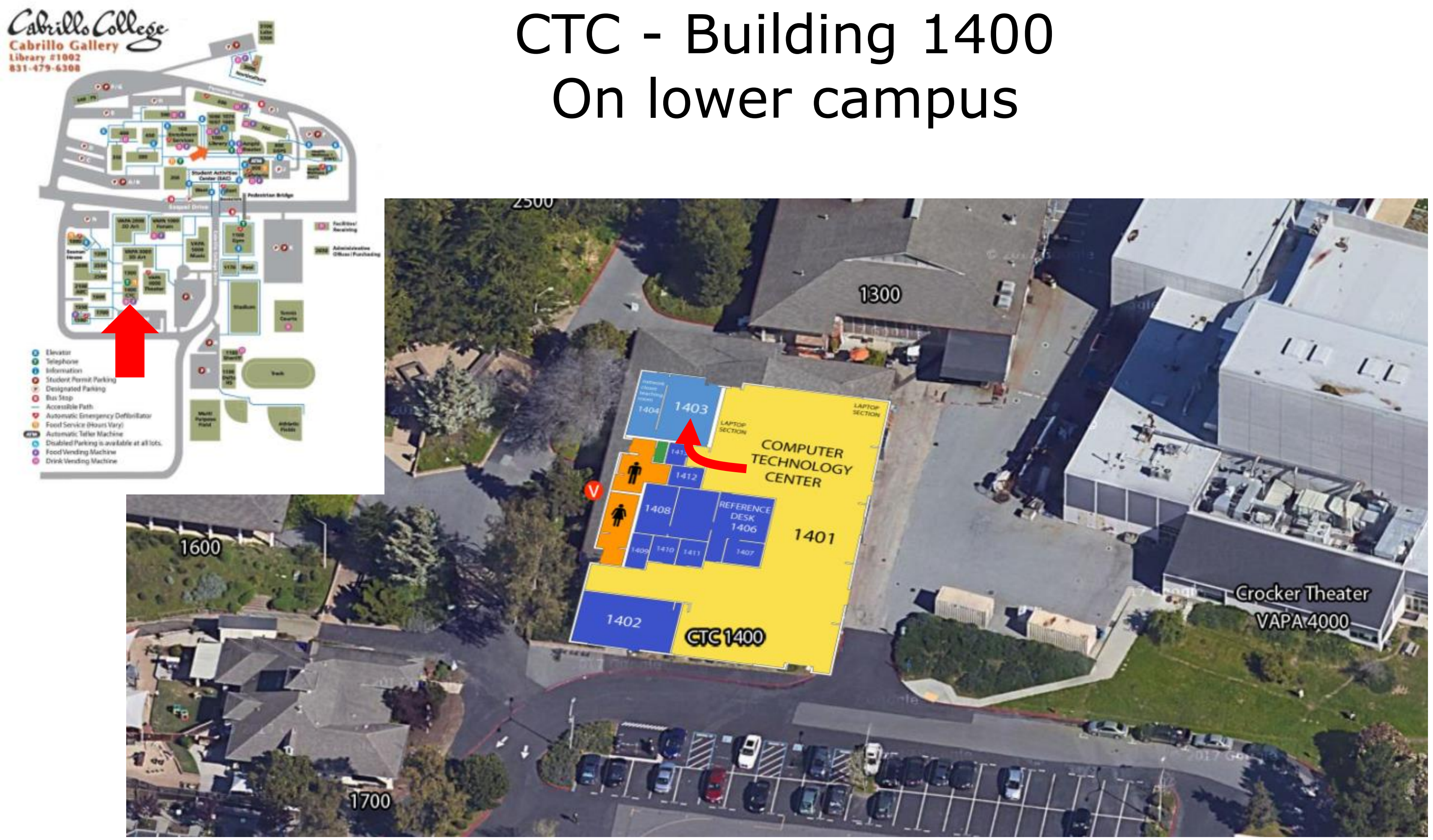
- Don't wait till the last minute to start.
- Plan for things to go wrong and give yourself time to ask questions and get answers.
- The *slower* you go the *sooner* you will be finished.
- A few minutes reading the forum can save you hour(s).
- Line up materials, references, equipment and software ahead 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.
- 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.

Getting Help When Stuck on an Assignment

- Google the topic/error message.
- Search the Lesson Slides (they are PDFs) for a relevant example on how to do something.
- Check the forum. *Someone else may have run into the same issue and found a way past it. If not start a new topic, 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:
  https://www.cabrillo.edu/salsa/listing.php?staffId=1426
  I'm in the CTC (room 1403) every Tuesday from 3:30-5:00 pm.
- 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!

CIS Labs always involve some troubleshooting!

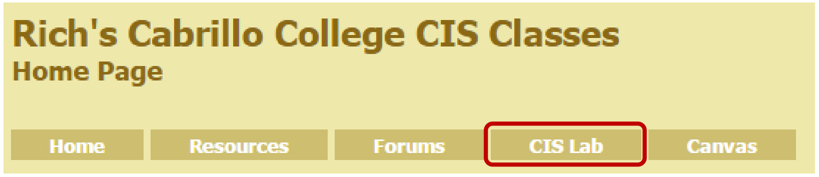
CTC - Building 1400
On lower campus



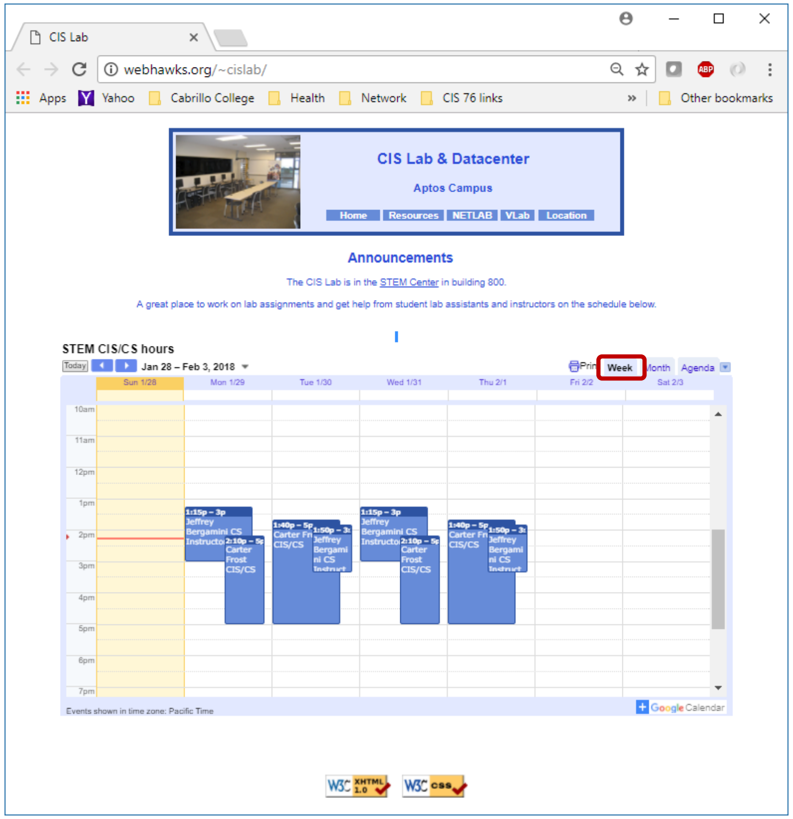
I will be in the CTC (room 1403) every Tuesday afternoon from 3-5:30

Help Available in the CIS Lab

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



Rich's Cabrillo College CIS Classes
Home Page



CIS Lab & Datacenter

Aptos Campus

Announcements

The CIS Lab is in the STEM Center in building 800.

A great place to work on lab assignments and get help from student lab assistants and instructors on the schedule below.

STEM CIS/CS hours

Jan 28 – Feb 3, 2018

	Sun 1/28	Mon 1/29	Tue 1/30	Wed 1/31	Thu 2/1	Fri 2/2	Sat 2/3
10am							
11am							
12pm							
1pm		1:15p – 3p Jeffrey Bergamini CS Instructo; 2:10p – 5p Carter Frost CIS/CS	1:40p – 5p Carter Fr CIS/CS; 1:50p – 3: Jeffrey Bergami ni CS Instruct	1:15p – 3p Jeffrey Bergamini CS Instructo; 2:10p – 5p Carter Frost CIS/CS	1:40p – 5p Carter Fr CIS/CS; 1:50p – 3: Jeffrey Bergami ni CS Instruct		
2pm							
3pm							
4pm							
5pm							
6pm							
7pm							

Events shown in time zone: Pacific Time

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

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 ...

5) If you had trouble doing the last test ...

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

Email: risimms@cabrillo.edu

Test 2 Post Mortem

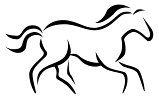
Test 2 – Results

Missed Q26 = 20
Missed Q30 = 17
Missed Q29 = 17
Missed Q25 = 17
Missed Q24 = 17
Missed Q23 = 14
Missed Q22 = 14
Missed Q21 = 14
Missed Q20 = 13
Missed Q17 = 13
Missed Q13 = 13
Missed Q4 = 12
Missed Q28 = 12
Missed Q2 = 12
Missed Q19 = 12

Missed Q18 = 12
Missed Q27 = 11
Missed Q9 = 10
Missed Q11 = 10
Missed Q12 = 7
Missed Q10 = 6
Missed Q16 = 5
Missed Q15 = 5
Missed Q7 = 4
Missed Q14 = 4
Missed Q8 = 3
Missed Q6 = 3
Missed Q5 = 1
Missed Q3 = 1
Missed Q1 = 1

Extra Credit
Missed Q33 = 21
Missed Q32 = 19
Missed Q31 = 17

For correct answers see test02.graded files in your home directory

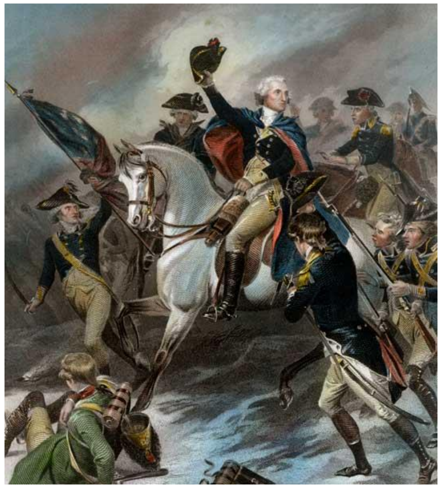


Q16) There is a file in the */etc* directory named *passwd*. This file has information on all user accounts including usernames, UIDs, first and last name, etc. What is the absolute pathname of this file?

Correct answer: /etc/passwd

http://www.sodahead.com/united-states/what-color-was-george-washingtons-white-horse/question-636725/

http://kids.britannica.com/comptons/art-55428/General-George-Washington-and-his-staff-welcoming-a-provision-train


http://www.mountvernon.org/content/revolutionary-war-princeton-white-horse

Housekeeping

war is not healthy for children and other living things
Coffee!
You can sleep when you're dead!
LATTE
ICE COLD
TOURIST MAP OF THE WORLD
New Zealand
We're Proud To Serve

Don't wait till midnight tonight to see if this worked! Submit with an earlier time.

1. Lab 8 due tonight

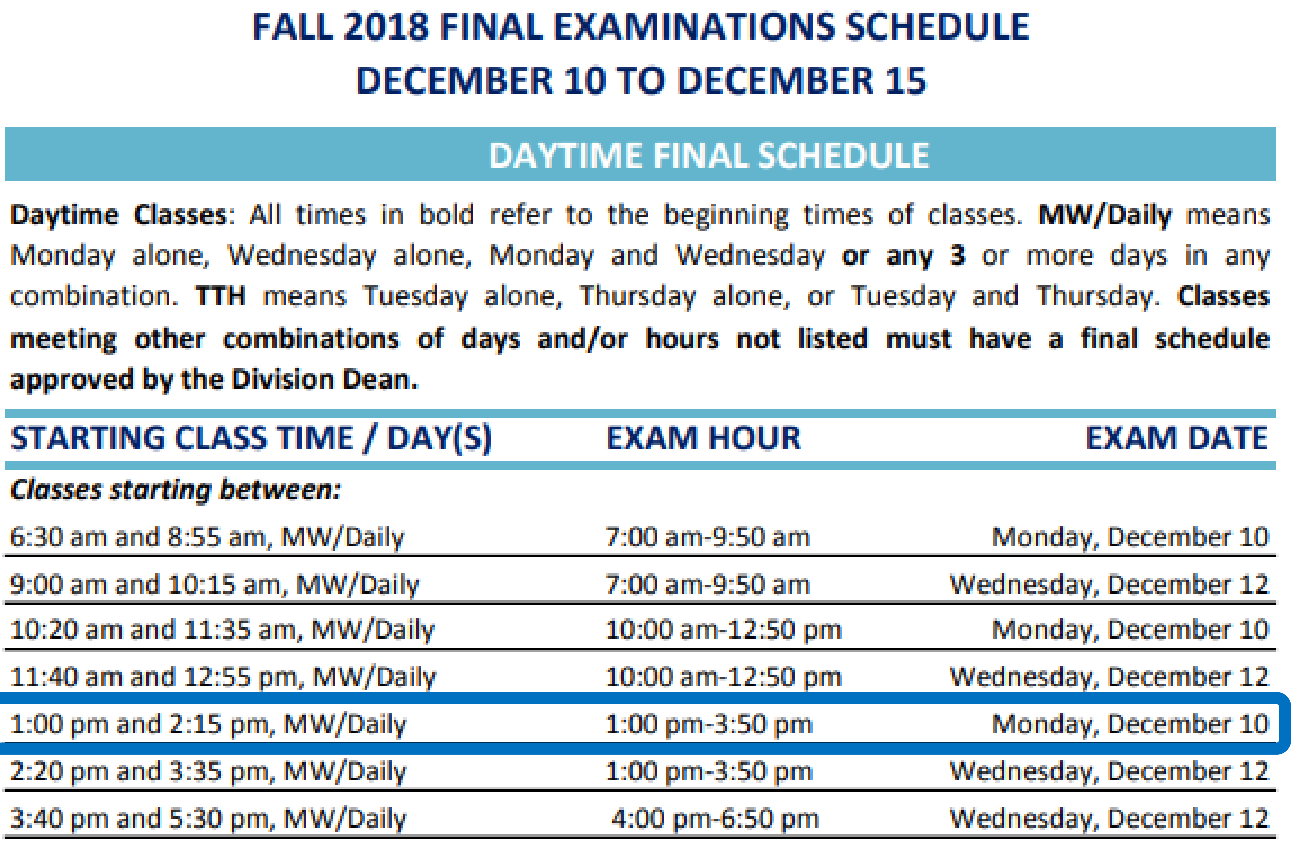
```
at 11:59pm
at> cat files.out bigshell > lab08
at> cp lab08 /home/rsimms/turnin/cis90/lab08.$LOGNAME
at> <Ctrl-D>
```

2. A **check8** script is available for Lab 8.

3. Read your email on Opus to verify your Lab 8 submission was received AND that you did not submit an empty file!

4. Note: Lab 9 and five posts due next week.

CIS 90 - Lesson 11

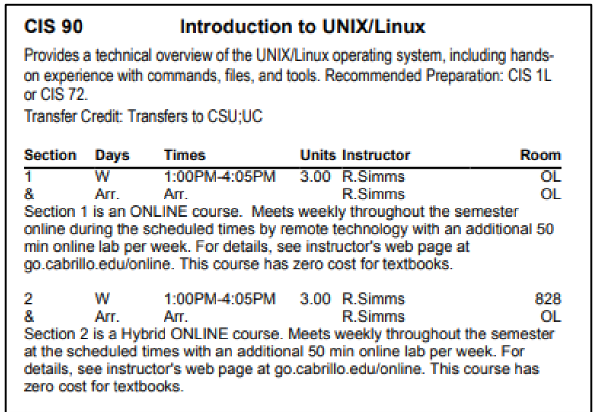


FALL 2018 FINAL EXAMINATIONS SCHEDULE
DECEMBER 10 TO DECEMBER 15

DAYTIME FINAL SCHEDULE

Daytime Classes: All times in bold refer to the beginning times of classes. **MW/Daily** means Monday alone, Wednesday alone, Monday and Wednesday **or any 3** or more days in any combination. **TTH** means Tuesday alone, Thursday alone, or Tuesday and Thursday. **Classes meeting other combinations of days and/or hours not listed must have a final schedule approved by the Division Dean.**

STARTING CLASS TIME / DAY(S)	EXAM HOUR	EXAM DATE
Classes starting between:		
6:30 am and 8:55 am, MW/Daily	7:00 am-9:50 am	Monday, December 10
9:00 am and 10:15 am, MW/Daily	7:00 am-9:50 am	Wednesday, December 12
10:20 am and 11:35 am, MW/Daily	10:00 am-12:50 pm	Monday, December 10
11:40 am and 12:55 pm, MW/Daily	10:00 am-12:50 pm	Wednesday, December 12
1:00 pm and 2:15 pm, MW/Daily	1:00 pm-3:50 pm	Monday, December 10
2:20 pm and 3:35 pm, MW/Daily	1:00 pm-3:50 pm	Wednesday, December 12
3:40 pm and 5:30 pm, MW/Daily	4:00 pm-6:50 pm	Wednesday, December 12



CIS 90 **Introduction to UNIX/Linux**

Provides a technical overview of the UNIX/Linux operating system, including hands-on experience with commands, files, and tools. Recommended Preparation: CIS 1L or CIS 72.
Transfer Credit: Transfers to CSU;UC

Section	Days	Times	Units	Instructor	Room
1	W	1:00PM-4:05PM	3.00	R.Simms	OL
&	Arr.	Arr.		R.Simms	OL

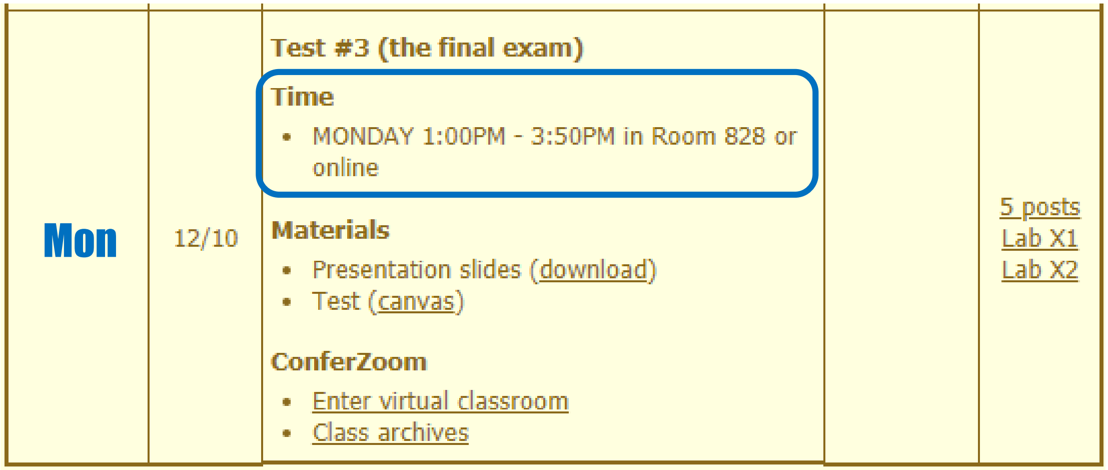
Section 1 is an ONLINE course. Meets weekly throughout the semester online during the scheduled times by remote technology with an additional 50 min online lab per week. For details, see instructor's web page at go.cabrillo.edu/online. This course has zero cost for textbooks.

Section	Days	Times	Units	Instructor	Room
2	W	1:00PM-4:05PM	3.00	R.Simms	828
&	Arr.	Arr.		R.Simms	OL

Section 2 is a Hybrid ONLINE course. Meets weekly throughout the semester at the scheduled times with an additional 50 min online lab per week. For details, see instructor's web page at go.cabrillo.edu/online. This course has zero cost for textbooks.

Heads up on Final Exam

Test #3 (final exam) is **MONDAY December 10th 1-3:50PM**



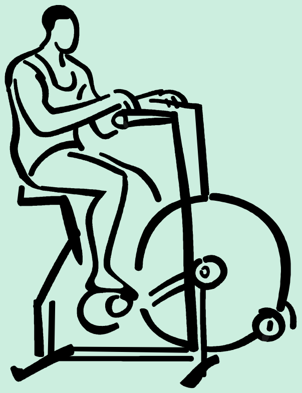
Mon	12/10	**Test #3 (the final exam)** **Time** • MONDAY 1:00PM - 3:50PM in Room 828 or online **Materials** • Presentation slides (download) • Test (canvas) **ConferZoom** • Enter virtual classroom • Class archives		5 posts Lab X1 Lab X2

Extra credit Labs X1/X2 and final posts ***due by 11:59PM***

Final grades *available by the end of the next day*

- All students will take the test at the same time. The test must be completed by **3:50PM**.
- Working and long distance students can take the test online via ConferZoom and Canvas.
- Working students will need to plan ahead to arrange time off from work for the test.
- Test #3 is **mandatory** (even if you have all the points you want)

grep workout



Some perfect times to use the **grep** command:

1) To search through the output of a command for some text

```
command | grep "text string"
```

2) To search inside one or more files for some text

```
grep "text string" file1 file2 … fileN
```

3) To search (recursively) inside all files in a branch of the UNIX file tree for some text

```
grep –R "text string" directory
```

grep usage – search output of a command

Is the CUPS daemon (print service) running right now?

```
/home/cis90/simben $ ps -ef | grep cups
root       1323      1  0 Jan21 ?        00:00:24 /usr/sbin/cupsd -f
simben90   6361   3202  0 11:26 pts/1    00:00:00 grep --color=auto cups
```

Yes it is, with PID=1323

grep practice

Is the cronjob daemon (**crond**) running right now?

If so, type the crond PID into the chat window

grep usage – search output of a command

Is the Apache web server (httpd) installed?

This shows all installed package names

This searches for package names containing "httpd"

```
/home/cis90/simben $ rpm -qa | grep httpd
httpd-tools-2.4.6-80.el7.centos.1.x86_64
httpd-2.4.6-80.el7.centos.1.x86_64
```

Yes, version 2.4.6 has been installed

```
/home/cis90/simben $ httpd -v
Server version: Apache/2.4.6 (CentOS)
Server built:   Jun 27 2018 13:48:59
```

grep practice

Which relational DBMS (Database Management System) is installed on Opus-II?

MySQL
PostgreSQL
MariaDB

Put the name and version in the chat window

FYI, this DBMS is used by the Forum

grep usage – search output of a command

When were the last 5 times I logged in?

```
/home/cis90/simben $ last | grep $LOGNAME | head -n5
simben90 pts/2        localhost        Sat Nov  3 16:00   still logged in
simben90 pts/6        2607:f380:80f:f8 Wed Oct 31 15:03 - 16:44  (01:41)
simben90 pts/6        2607:f380:80f:f8 Wed Oct 31 12:32 - 15:03  (02:30)
simben90 pts/2        c-73-222-184-235 Tue Oct 30 12:54 - 15:09  (02:15)
simben90 pts/0        c-73-222-184-235 Tue Oct 30 12:53 - 14:14  (01:21)
/home/cis90/simben $
```

This scans the latest wtmp log file and lists your most recent five logins to Opus-II

grep practice

For the time period covered by the current wtmp log file. What was the date of your earliest login?

Type your earliest login date into the chat window

grep usage – search output of a command

```
[rsimms@oslab ~]$ ls /bin/{bash,sh,ksh,csh,tcsh}
/bin/bash  /bin/csh  /bin/ksh  /bin/sh  /bin/tcsh

[rsimms@oslab ~]$ ksh
$ sh
sh-4.2$ csh
```

Look familiar? (lab 8) Shows how to compare shells by size and record the biggest one in a file.

size

```
[rsimms@oslab ~]$ ps -l
F S   UID   PID  PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
4 S  1201  9483  9476  0  80   0 - 28881 do_wai pts/1    00:00:00 bash
0 S  1201  9533  9483  0  80   0 - 29280 do_wai pts/1    00:00:00 ksh
0 S  1201  9557  9533  0  80   0 - 28847 do_wai pts/1    00:00:00 sh
0 S  1201  9561  9557  0  80   0 - 29876 sigsus pts/1    00:00:00 csh
0 R  1201  9771  9561  0  80   0 - 37235 -      pts/1    00:00:00 ps

[rsimms@oslab ~]$ ps -l | grep csh
0 S  1201  9561  9557  0  80   0 - 29876 sigsus pts/1    00:00:00 csh

[rsimms@oslab ~]$ ps -l | grep csh > bigshell

[rsimms@oslab ~]$ cat bigshell
0 S  1201  9561  9557  0  80   0 - 29876 sigsus pts/1    00:00:00 csh
```

grep practice

Instructor note:
Login directly to simben90 (don't su)
Give write permission to others on Benji's terminal device: ***chmod o+w $(tty)***

- Run **bash**, **ksh**, **sh** and **csh** shells and use **ps -l** to see which is the smallest.
- Redirect the line of **ps -l** output for the smallest shell to Benji Simms's terminal: **/dev/pts/??**
- Sign it with **echo "From *first name*" > /dev/pts/??**
- Then **exit** each shell till your are back to just one bash shell running.

grep usage – search inside files

How many CIS 90 user accounts are there?

```
/home/cis90/simben $ grep :1090: /etc/passwd | wc -l
43

/home/cis90/simben $ grep cis90 /etc/passwd | wc -l
43

/home/cis90/simben $ grep "^.*90" /etc/passwd | wc -l
43
```

FYI only

There are 43

The third example is a "regular expression". For more information see the Resources page of the website.

grep practice

How many CIS 76 accounts are there on Opus-II?

Type the number of CIS 76 accounts into the chat window

grep usage – search inside files

Example: What is my account information in /etc/passwd?

```
/home/cis90/simben $ grep $LOGNAME /etc/passwd
simben90:x:1000:90:Benji Simms:/home/cis90/simben:/bin/bash
```

or

```
/home/cis90/simben $ grep simben90 /etc/passwd
simben90:x:1000:90:Benji Simms:/home/cis90/simben:/bin/bash
```

or

```
/home/cis90/simben $ cat /etc/passwd | grep $LOGNAME
simben90:x:1000:90:Benji Simms:/home/cis90/simben:/bin/bash
```

username

Comment

Home directory

Shell

Group ID (GID)

User ID (UID)

password (just a placeholder now)

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

grep practice

Does your user ID in */etc/passwd* match the uid output by the **id** command?

*Type your answer (yes or no) and your uid from the **id** command into the chat window*

grep usage – search inside files in all or part of the file tree

All the system configuration files are in the /etc directory

Where does the system set your "prompt" variable?

```
/home/cis90/simben $ grep -r "PS1=" /etc 2> /dev/null
/etc/bashrc:   [ "$PS1" = "\\s-\\v\\\$ " ] && PS1="[\u@\h \W]\\$ "
/etc/bashrc:   #   PS1="[\u@\h:\l \W]\\$ "
```

It is set more than once during login. We will learn in a future lesson that the one in .bash_profile is done last and is what you end up using.

```
/home/cis90/simben $ grep PS1= .bash_profile
PS1='$PWD $ '
```

grep practice

Find the file in the */usr/share* branch of the file tree that contains "playing hot potato".

Type the absolute pathname of the file in the chat window.

Shell
Six Steps
(REVIEW)

Activity

This is Benji's home directory

```
/home/cis90/simben $ ls -F
badevents     Directory3/  f2.graded          lab09           mylog        text.err          what_am_i
bag/          dogs/        Hidden/            Lab2.0/         Poems/       text.fxd          words
bigfile       dogs.tar     jobs/              Lab2.1/         proposal1    timecal*
bigfile.bak   dulces@      lab01.graded       labx2           proposal2    trash
bin/          dups         lab02-collection   letter          proposal3    treat1
candy         edits/       lab04.graded       log             small_town   uhistory
cis90_html/   empty        lab04-mydata       Miscellaneous/  spellk       uhistory.bak
dead.letter   f1.graded    lab07              mission         sweets       uhistory.rsimms
```

Benji wants to find some treats and types this command

```
/home/cis90/simben $ find / -name treat* 2> /dev/null
```

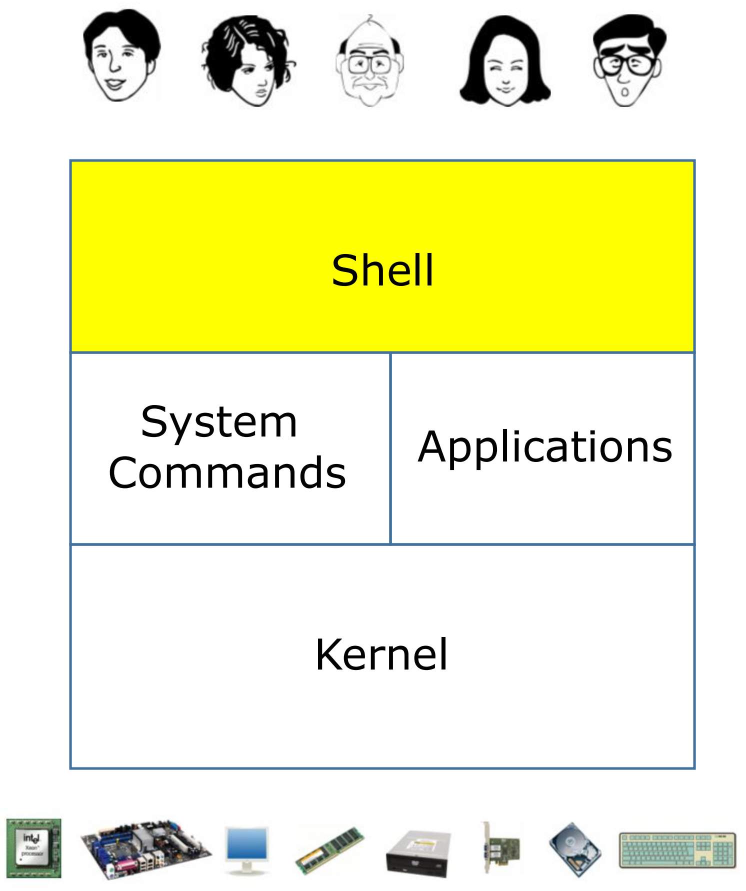
Write what you think will happen in the chat window

Example Command

```
/home/cis90/simben $ find / -name treat* 2> /dev/null
/home/cis90/watshe/treat1
/home/cis90/seasky/treat1
/home/cis90/simben/treat1
/home/cis90/milhom/treat1
/home/cis90/rodduk/treat1
/home/cis90/berale/bag/treat1
/home/cis90/cireri/treat1
/home/cis90/espdom/treat1
/home/cis90/espdom/bag/treat1
/home/cis90/evabla/treat1
/home/cis90/farton/bag/treat1
< snipped >
/home/cis90/pindan/bag/treat1
/home/cis90/siecar/treat1
/home/cis90/steisa/treat1
/home/cis90/vasmig/treat1
/home/cis90/caljos/treat1
/home/cis90/gongab/treat1
/home/cis90/learya/treat1
/home/cis90/lewali/bag/treat1
/home/cis90/rojfre/treat1
/home/cis90/rojfre/bag/treat1
/home/cis90/serjan/bag/treat1
/home/cis90/alvjon/bag/treat1
/home/cis90/simben $
```

Note: Benji has a file named treat1 in his home directory

Prompt Step



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

Prompt Step
(uses PS1 variable)

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

bash using your PS1 variable creates and outputs your prompt which is written to your terminal device

- Benji is using the bash shell. There are many other shells such as sh, ksh and csh. In */etc/passwd* the last field in the line for his account determines the shell that is run when logging in.

```
/home/cis90/simben $ grep $LOGNAME /etc/passwd
simben90:x:1001:190:Benji Simms:/home/cis90/simben:/bin/bash
```

- The bash program resides in the */bin* directory.

```
/home/cis90/simben $ ls -l /bin/bash
-rwxr-xr-x. 1 root root 874248 May 10  2012 /bin/bash
```

- The command prompt appearance is defined by the PS1 variable. You can output a prompt yourself using **echo $PS1**

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

Prompt Step

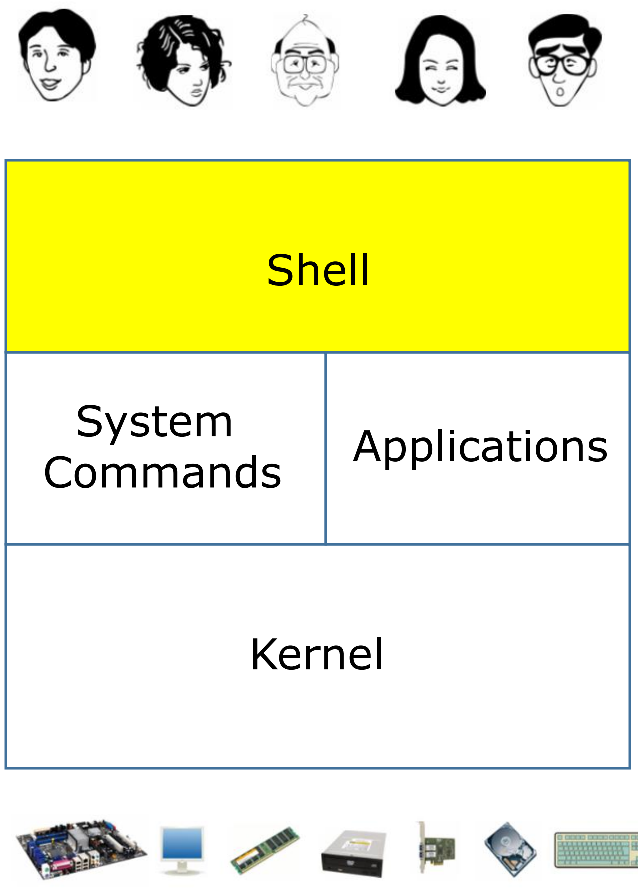
Note there is an invisible <newline> metacharacter at the end of the command

```
/home/cis90/simben $ find / -name treat* 2> /dev/null
```

Benji types this find command in response to the shell prompt

The prompt step is not complete until the user presses the Enter/Return key

Parse Step



1) Prompt

2) Parse

3) Search

4) Execute

5) Nap

6) Repeat

Parse Step

The shell uses spaces to separate options, arguments and redirection

```
find / -name treat* 2> /dev/null
```

The shell must expand filename expansion characters and variables during the parse step.

Parsing RESULTS:

Command: **find**

Options and arguments:
  /
  -name
  treat1

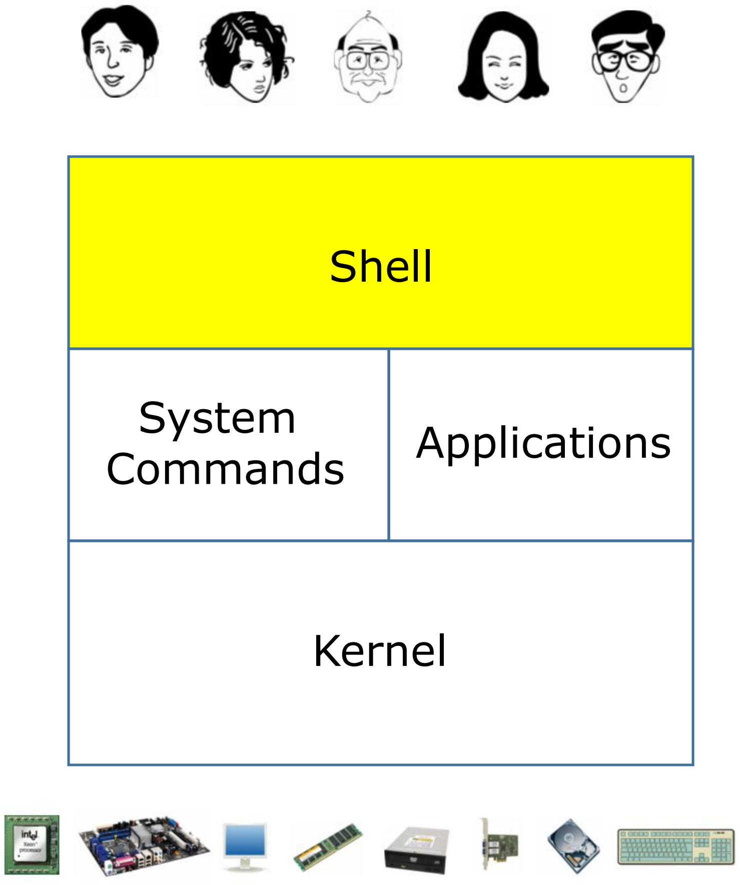
This will be passed to the command (if the command can be located on the path)

Redirection:
  Connect **stderr** to **/dev/null** (the "bit bucket")

This will be handled by the shell. The command, if loaded, will not see this

Note: Because Benji had a treat1 file in his home directory, the shell expands treat to treat1*

Search Step



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

Search Step

(uses PATH variable)

Command: **find**

*The shell now must search, in order, every directory on Benji's path to locate the first occurrence of the **find** command.*

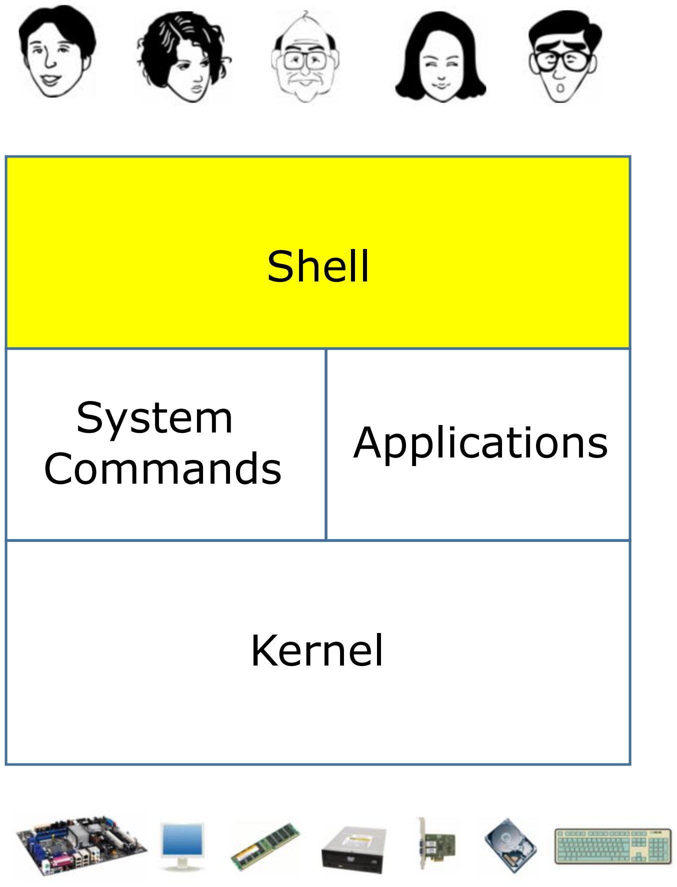
Benji's path is defined by the value of his PATH variable

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

The shell locates the find command in the /usr/bin directory

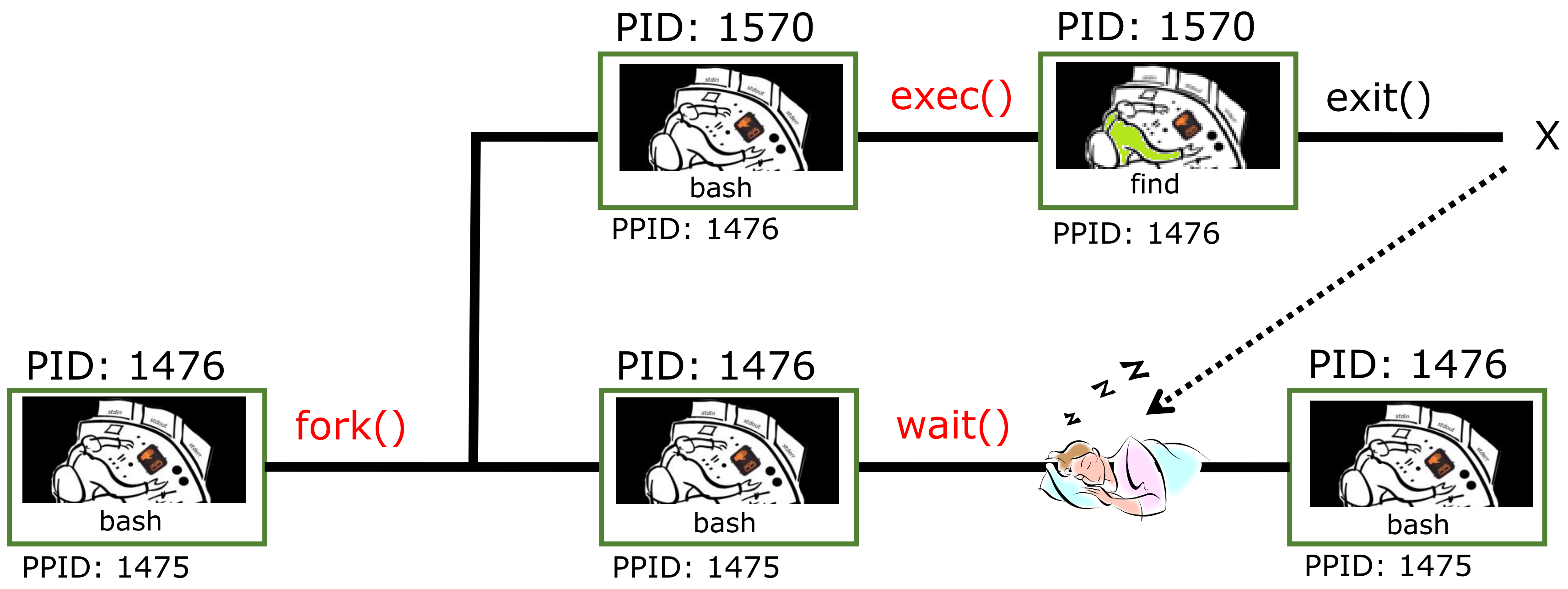
```
/home/cis90/simben $ echo $PATH
/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/home/cis90/simben/../bin:/home/cis90/simben/bin:.
/home/cis90/simben $ type find
find is /usr/bin/find
```

Execute Step



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

Execute Step

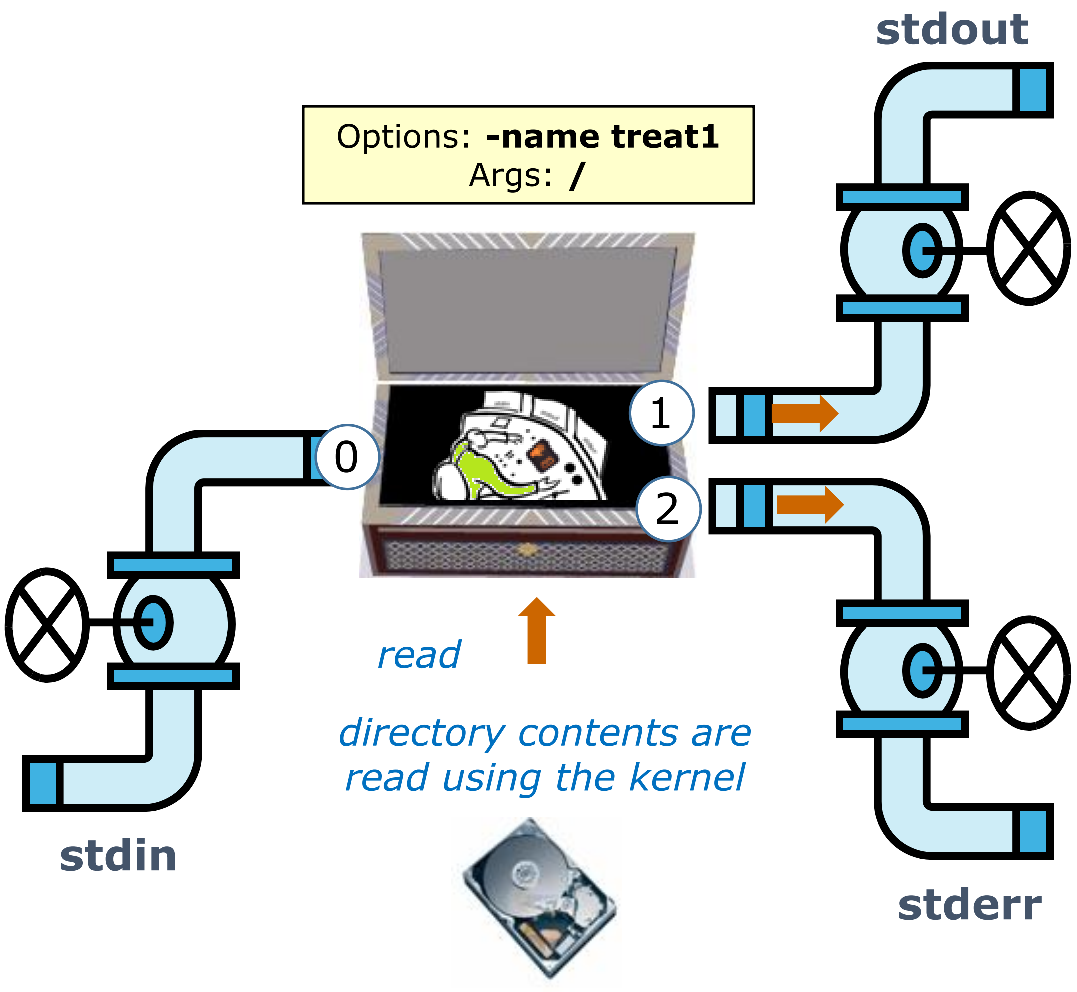


bash executes the **find** command by:

1) Cloning itself with a **fork()** system call to create a new child process.
2) With an **exec()** system call, the new child process is overlaid with the **find** code instructions.
3) bash sleeps by making a **wait()** system call while the find child process runs.
4) The child process makes an **exit()** system call when it has finished.
5) After that, the parent bash process wakes up and the child process is killed.

Execute Step

```
/home/cis90/simben $ find / -name treat* 2> /dev/null
```


Options: **-name treat1**
Args: **/**
stdout
1
0
2
stdin
stderr
read
directory contents are read using the kernel

```
/home/cis90/primic/treat1
/home/cis90/juetay/treat1
/home/cis90/porjos/treat1
/home/cis90/beycha/bag/treat1
/home/cis90/drydan/bag/treat1
/home/cis90/rodduk/treat1
/home/cis90/tosbre/treat1
/home/cis90/remlis/treat1
/home/cis90/linmay/treat1
/home/cis90/brevic/treat1
< snipped >
```

/dev/null

```
find: `/lost+found': Permission denied
find: `/var/empty/sshd': Permission denied
find: `/var/log/sssd': Permission denied
< snipped >
```

This is what the find process might look like

stdin
stdout
stderr
/
-name
treat1

*The **find** process is running*

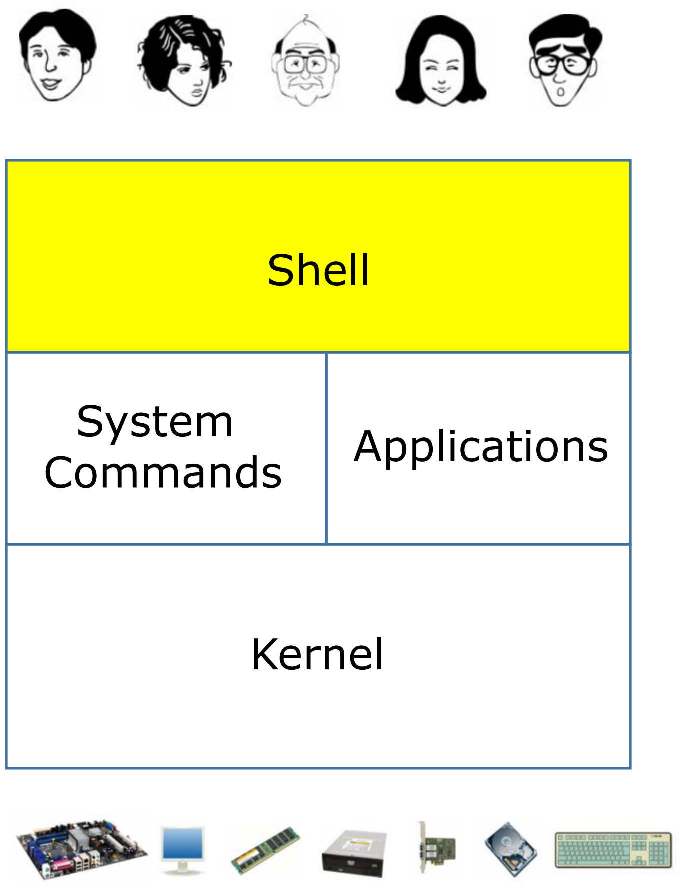
A **process:**

- Is provided with parsed/expanded options and arguments from the shell

- may read from **stdin**
- may write to **stdout**
- may write error messages to **stderr**

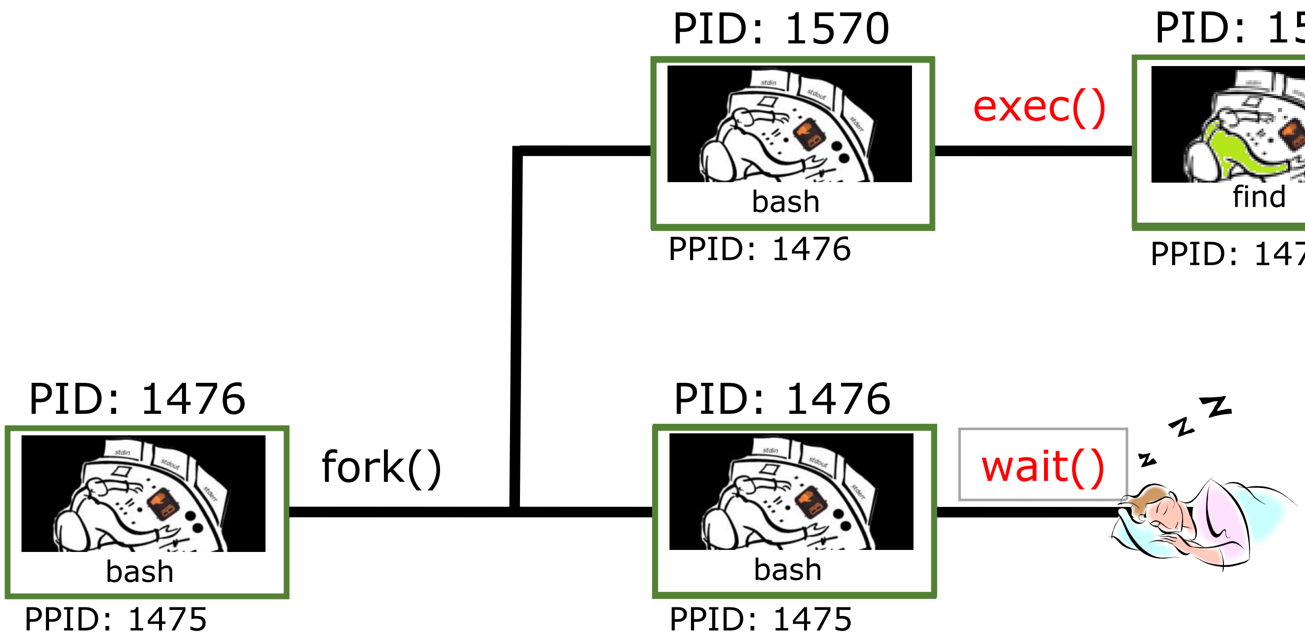
- and may get interrupted from time to time by a **signal**

Nap Step



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

Nap Step



*The PS command shows Benji's **find** command is running as a child process while the parent bash shell sleeps*

Sleeping

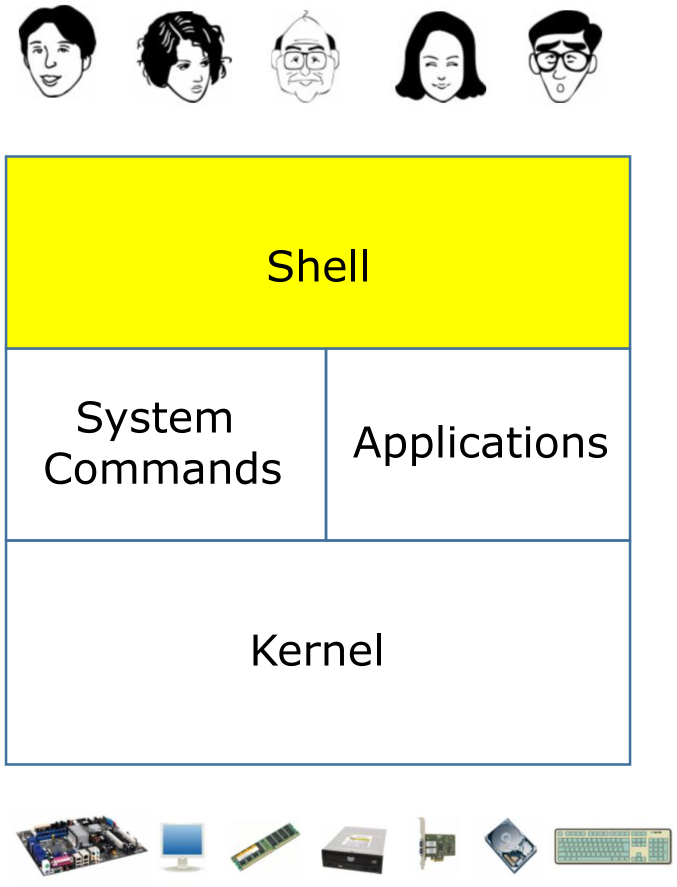
```
[rsimms@oslab ~]$ ps -l -u simben90
F S   UID   PID  PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
5 S  1001  1475  1470  0  80   0 -  3392 ?      ?        00:00:00 sshd
0 S  1001  1476  1475  0  80   0 -  1308 ?      pts/1    00:00:00 bash
0 R  1001  1570  1476 40  80   0 -  1179 ?      pts/1    00:00:00 find
```

Parent

Child

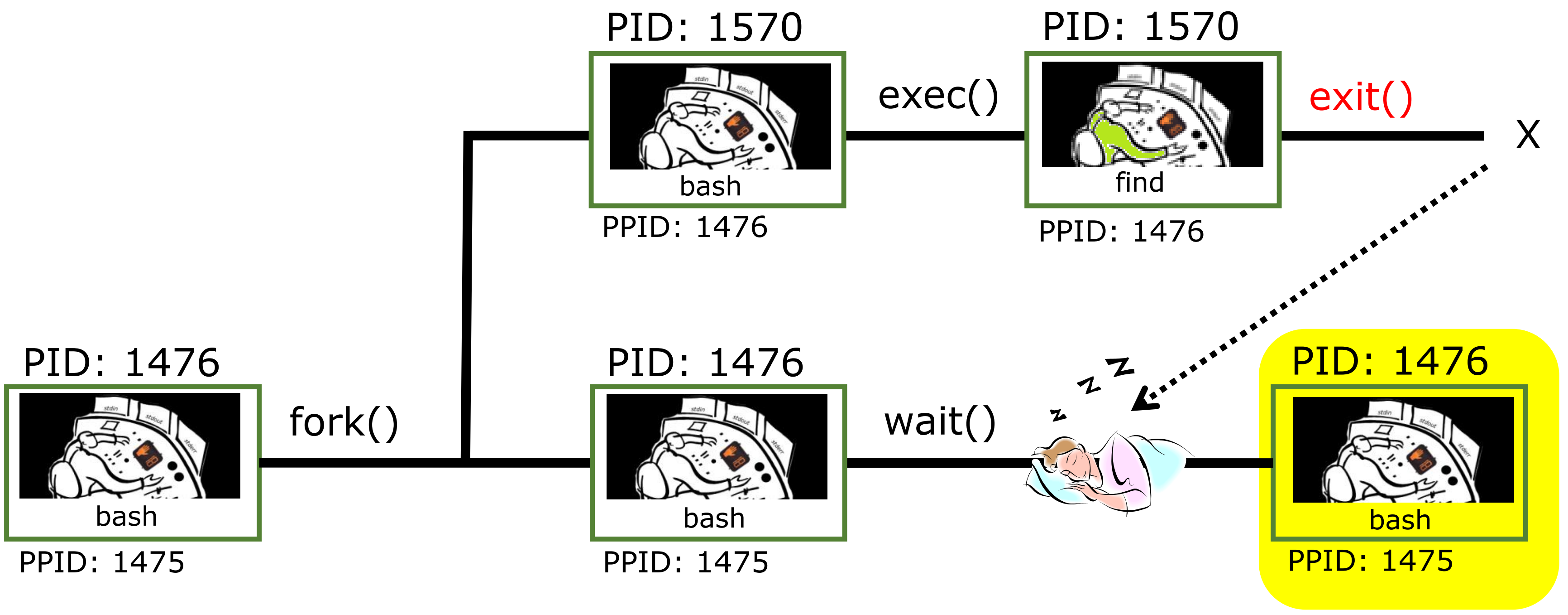
Running

Repeat Step



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

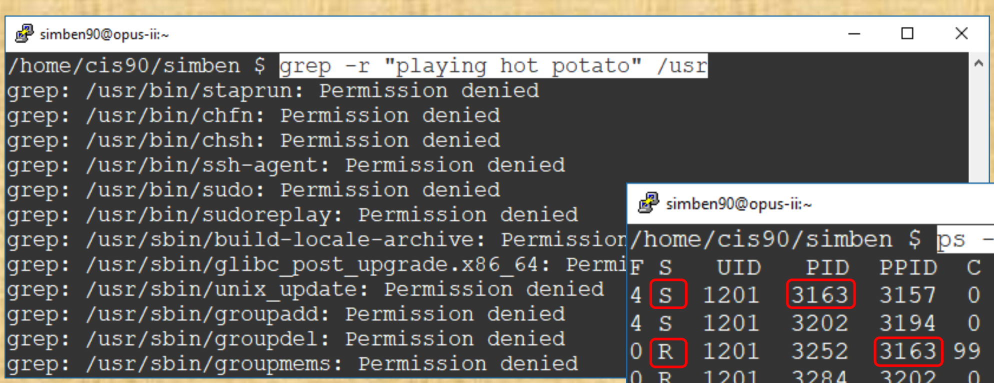
Repeat Step



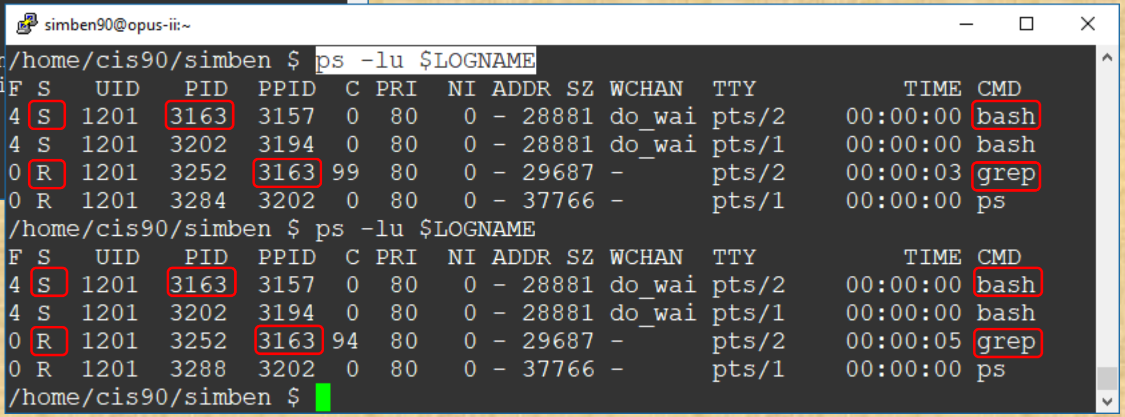
*The child process makes an **exit()** system call when it has finished. The parent bash process wakes up, the child process is killed and we are ready to start the process all over again with the next command.*

Process activity

- Start a second login session and see if you can illustrate the parent sleeping while a child runs.
- In one session run: **`grep -r "playing hot potato" /usr`**
- In the second session use repeatedly: **`ps -lu $LOGNAME`**
- The **ps** output should show "parent" bash S=Sleeping while the "child" **grep** command is either R=Running or in D=Uninterruptible sleep (IO)



```
/home/cis90/simben $ grep -r "playing hot potato" /usr
grep: /usr/bin/staprun: Permission denied
grep: /usr/bin/chfn: Permission denied
grep: /usr/bin/chsh: Permission denied
grep: /usr/bin/ssh-agent: Permission denied
grep: /usr/bin/sudo: Permission denied
grep: /usr/bin/sudoreplay: Permission denied
grep: /usr/sbin/build-locale-archive: Permission
grep: /usr/sbin/glibc_post_upgrade.x86_64: Permi
grep: /usr/sbin/unix_update: Permission denied
grep: /usr/sbin/groupadd: Permission denied
grep: /usr/sbin/groupdel: Permission denied
grep: /usr/sbin/groupmems: Permission denied
```



```
/home/cis90/simben $ ps -lu $LOGNAME
F S   UID   PID  PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
4 S  1201  3163  3157  0  80   0 - 28881 do_wai pts/2    00:00:00 bash
4 S  1201  3202  3194  0  80   0 - 28881 do_wai pts/1    00:00:00 bash
0 R  1201  3252  3163 99  80   0 - 29687 -      pts/2    00:00:03 grep
0 R  1201  3284  3202  0  80   0 - 37766 -      pts/1    00:00:00 ps
/home/cis90/simben $ ps -lu $LOGNAME
F S   UID   PID  PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
4 S  1201  3163  3157  0  80   0 - 28881 do_wai pts/2    00:00:00 bash
4 S  1201  3202  3194  0  80   0 - 28881 do_wai pts/1    00:00:00 bash
0 R  1201  3252  3163 94  80   0 - 29687 -      pts/2    00:00:05 grep
0 R  1201  3288  3202  0  80   0 - 37766 -      pts/1    00:00:00 ps
/home/cis90/simben $
```

Write your parent bash status and PID into the chat window

bash, pid=3163, status=S (sleeping)

Signals (Review)

Signals

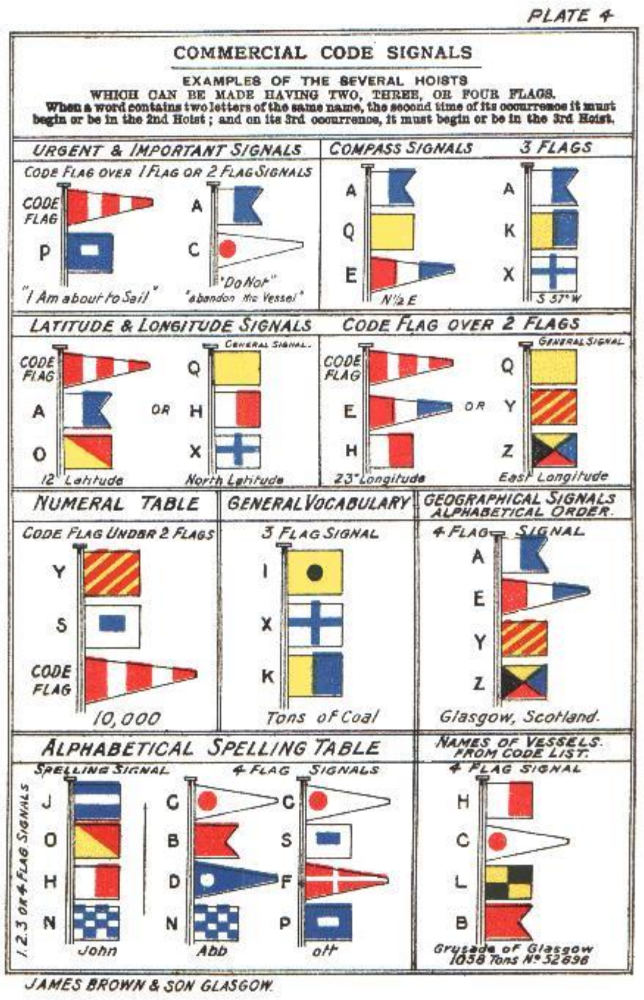
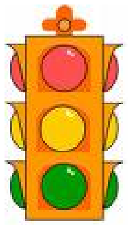
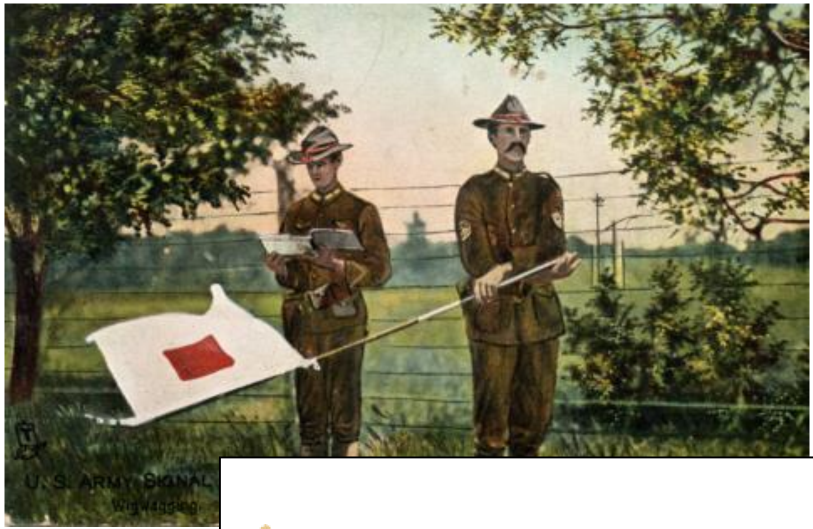
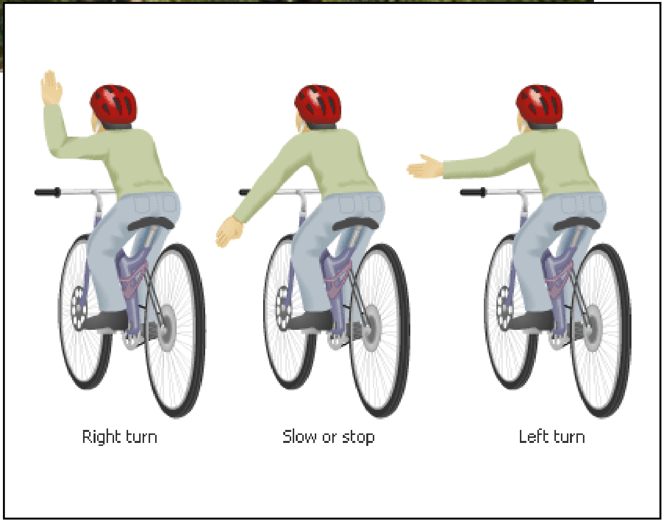


PLATE 4
COMMERCIAL CODE SIGNALS
EXAMPLES OF THE SEVERAL HOISTS
WHICH CAN BE MADE HAVING TWO, THREE, OR FOUR FLAGS.
When a word contains two letters of the same name, the second time of its occurrence it must begin or be in the 2nd Hoist; and on its 3rd occurrence, it must begin or be in the 3rd Hoist.
URGENT & IMPORTANT SIGNALS
CODE FLAG OVER 1 FLAG OR 2 FLAG SIGNALS
"I Am about to Sail"
"Do Not abandon the Vessel"
COMPASS SIGNALS
3 FLAGS
N½E
S 57° W
LATITUDE & LONGITUDE SIGNALS
CODE FLAG OVER 2 FLAGS
12° Latitude
North Latitude
23° Longitude
East Longitude
NUMERAL TABLE
CODE FLAG UNDER 2 FLAGS
10,000
GENERAL VOCABULARY
3 FLAG SIGNAL
Tons of Coal
GEOGRAPHICAL SIGNALS ALPHABETICAL ORDER.
4 FLAG SIGNAL
Glasgow, Scotland.
ALPHABETICAL SPELLING TABLE
SPELLING SIGNAL
4 FLAG SIGNALS
1.2.3 OR 4 FLAG SIGNALS
John
Abb
off
NAMES OF VESSELS FROM CODE LIST.
4 FLAG SIGNAL
Grusade of Glasgow 1058 Tons Nº 32898
JAMES BROWN & SON GLASGOW




U. S. ARMY SIGNAL
Wig-wagging


Right turn
Slow or stop
Left turn

This is what a process might look like

stdin
stdout
stderr

A **process:**

- Is provided with parsed/expanded options and arguments from the shell

- may read from **stdin**
- may write to **stdout**
- may write error messages to **stderr**

- and may get interrupted from time to time by a **signal**

*A **process** is a **program** that has been loaded into memory and is either running (executing instructions) or waiting to run*

Signals

The result of sending a signal to a process:

- be ignored
- default action (die)
- execute some predefined function

This running process gets signal 20 (SIGTSTP)

stdin
stdout
stderr
20

Signals

SIGHUP	1	Hangup (POSIX)	
SIGINT	2	Terminal interrupt (ANSI)	**Ctrl-C**
SIGQUIT	3	Terminal quit (POSIX)	**Ctrl-\**
SIGILL	4	Illegal instruction (ANSI)	
SIGTRAP	5	Trace trap (POSIX)	
SIGIOT	6	IOT Trap (4.2 BSD)	
SIGBUS	7	BUS error (4.2 BSD)	
SIGFPE	8	Floating point exception (ANSI)	
SIGKILL	9	Kill (can't be caught or ignored) (POSIX)	
SIGUSR1	10	User defined signal 1 (POSIX)	
SIGSEGV	11	Invalid memory segment access (ANSI)	
SIGUSR2	12	User defined signal 2 (POSIX)	
SIGPIPE	13	Write on a pipe with no reader, Broken pipe (POSIX)	
SIGALRM	14	Alarm clock (POSIX)	
SIGTERM	15	Termination (ANSI)	

Use kill –l to see all signals

Signals

SIGSTKFLT	16	Stack fault
SIGCHLD	17	Child process has stopped or exited, changed (POSIX)
SIGCONT	18	Continue executing, if stopped (POSIX)
SIGSTOP	19	Stop executing(can't be caught or ignored) (POSIX)
SIGTSTP	20	Terminal stop signal (POSIX) **Ctrl-Z or Ctrl-F**
SIGTTIN	21	Background process trying to read, from TTY (POSIX)
SIGTTOU	22	Background process trying to write, to TTY (POSIX)
SIGURG	23	Urgent condition on socket (4.2 BSD)
SIGXCPU	24	CPU limit exceeded (4.2 BSD)
SIGXFSZ	25	File size limit exceeded (4.2 BSD)
SIGVTALRM	26	Virtual alarm clock (4.2 BSD)
SIGPROF	27	Profiling alarm clock (4.2 BSD)
SIGWINCH	28	Window size change (4.3 BSD, Sun)
SIGIO	29	I/O now possible (4.2 BSD)
SIGPWR	30	Power failure restart (System V)

Use kill –l to see all signals

Signals

Signals are asynchronous messages sent to processes

They can result in one of three courses of action:

1. be ignored,
2. default action (die)
3. execute some predefined function.

Signals are sent:

kill command

Using the kill command: **$ kill -# PID**

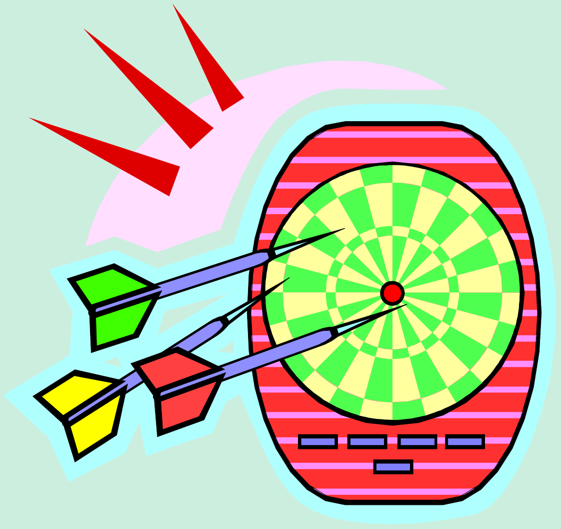
- Where # is the signal number and PID is the process id.
- if no number is specified, SIGTERM (-15) is sent.

Using special keystrokes

- limited to just a few signals
- limited to when you have control of the keyboard

Use kill –l to see all signals

Target Practice



Activity

1) Run the **annoy** program

2) Try sending it a SIGINT with **Ctrl-C**

3) Try sending it a SIGQUIT with **Ctrl-\**

4) Bring up another terminal and try signals 1 through 64
   - Use **ps –u $LOGNAME** to find the **annoy** *PID*

   - Try **kill -1 *PID***
   - Try **kill -2 *PID***
   - Try **kill -3 *PID***
   - *and so forth …*

   OR

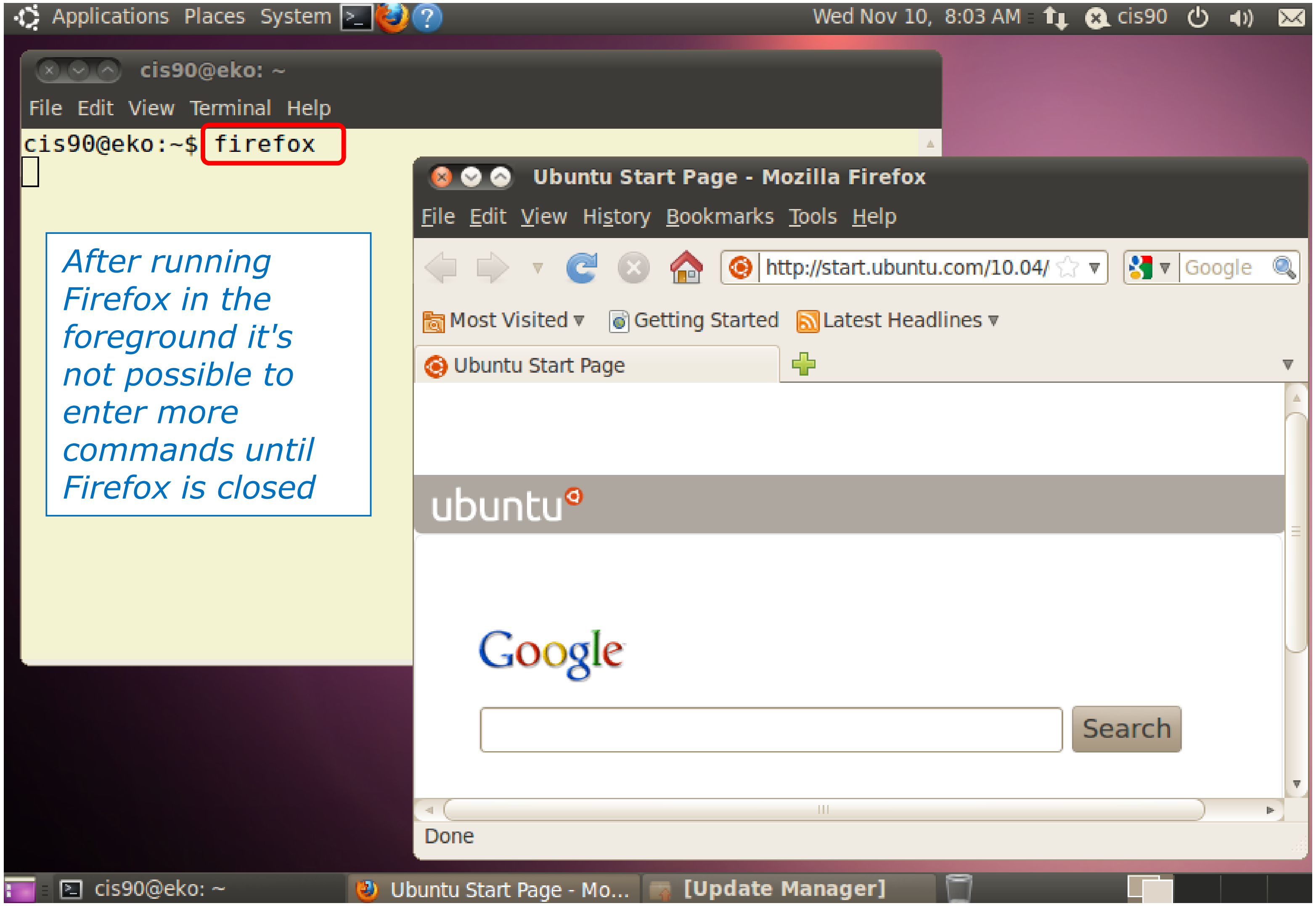
   - Try **killall -1 annoy**
   - Try **killall -2 annoy**
   - Try **killall -3 annoy**
   - *and so forth …*

*Write the signals that kill **annoy** into the chat window*

Using &

to run a command in the background

Job Control
Using **&** to run a command in the background

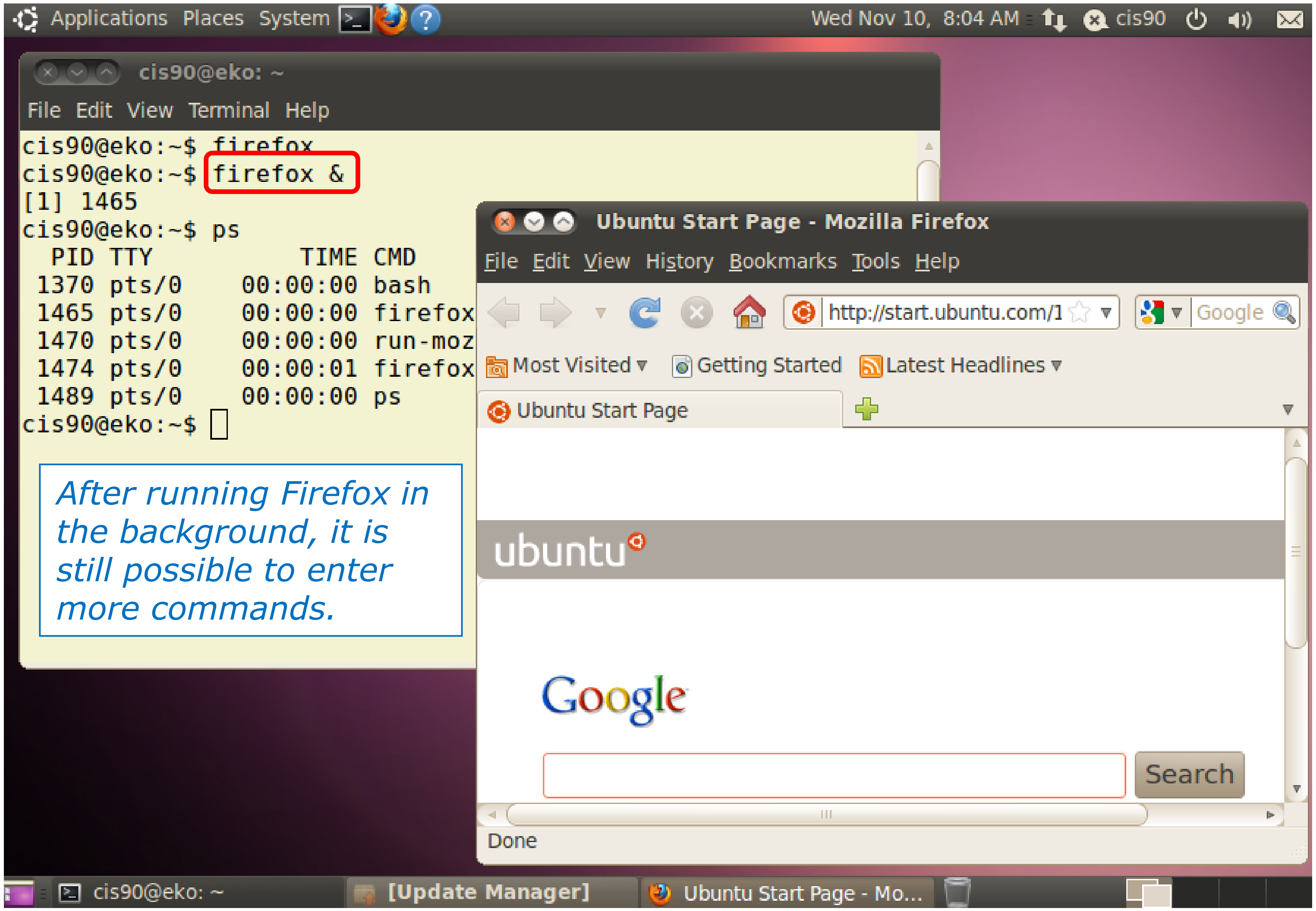

Applications Places System
Wed Nov 10, 8:03 AM
cis90
cis90@eko: ~
File Edit View Terminal Help
Ubuntu Start Page - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://start.ubuntu.com/10.04/
Google
Most Visited
Getting Started
Latest Headlines
Ubuntu Start Page
ubuntu
Google
Search
Done
cis90@eko: ~
Ubuntu Start Page - Mo...
[Update Manager]

```
cis90@eko:~$ firefox
```

After running Firefox in the foreground it's not possible to enter more commands until Firefox is closed

Job Control

Using **&** to run a command in the background



```
cis90@eko:~$ firefox
cis90@eko:~$ firefox &
[1] 1465
cis90@eko:~$ ps
  PID TTY          TIME CMD
 1370 pts/0    00:00:00 bash
 1465 pts/0    00:00:00 firefox
 1470 pts/0    00:00:00 run-moz
 1474 pts/0    00:00:01 firefox
 1489 pts/0    00:00:00 ps
cis90@eko:~$
```

After running Firefox in the background, it is still possible to enter more commands.

Applications Places System
Wed Nov 10, 8:04 AM
cis90
cis90@eko: ~
File Edit View Terminal Help
Ubuntu Start Page - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://start.ubuntu.com/1
Google
Most Visited
Getting Started
Latest Headlines
Ubuntu Start Page
ubuntu
Google
Search
Done
cis90@eko: ~
[Update Manager]
Ubuntu Start Page - Mo...

& append to a command to run it in the background

Example 1

```
/home/cis90/simben $ grep -r "playing hot potato" /usr 2> /dev/null
```

No prompt, bash is asleep.

For long running commands or scripts you must wait for the command to finish before you type more commands

Example 2

```
/home/cis90/simben $ grep -r "playing hot potato" /usr /opt 2> /dev/null &
[1] 7921
/home/cis90/simben $ date
Fri Apr 13 13:44:00 PDT 2018
```

Hit enter to get the prompt and continue working while the find command runs in the background

Job Control (Review)

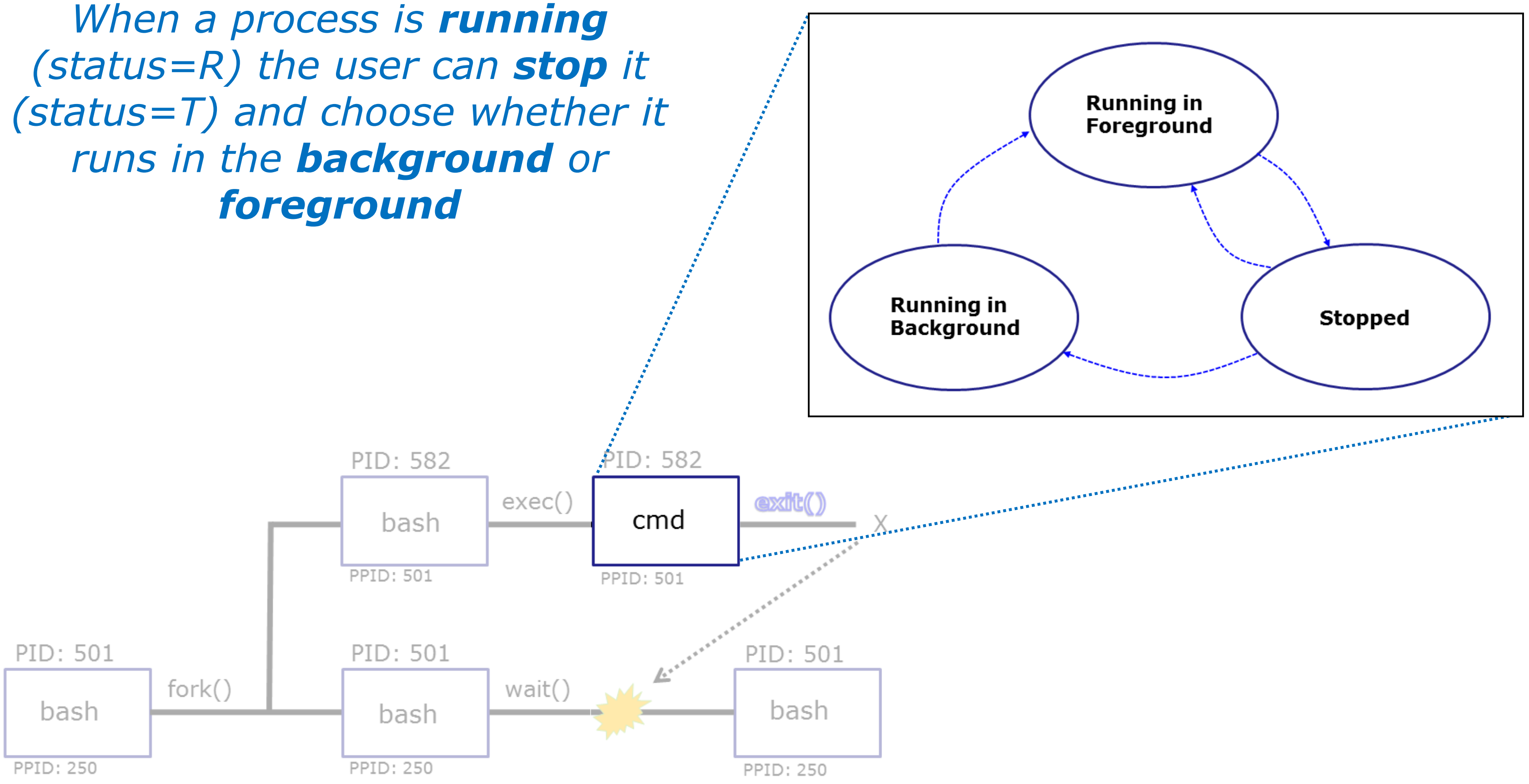
Job Control
A feature of the bash shell

&	Append to a command to run it in the background
bg	Resumes a suspended job in the background
fg	Brings the most recent background process to the foreground
jobs	Lists all background jobs

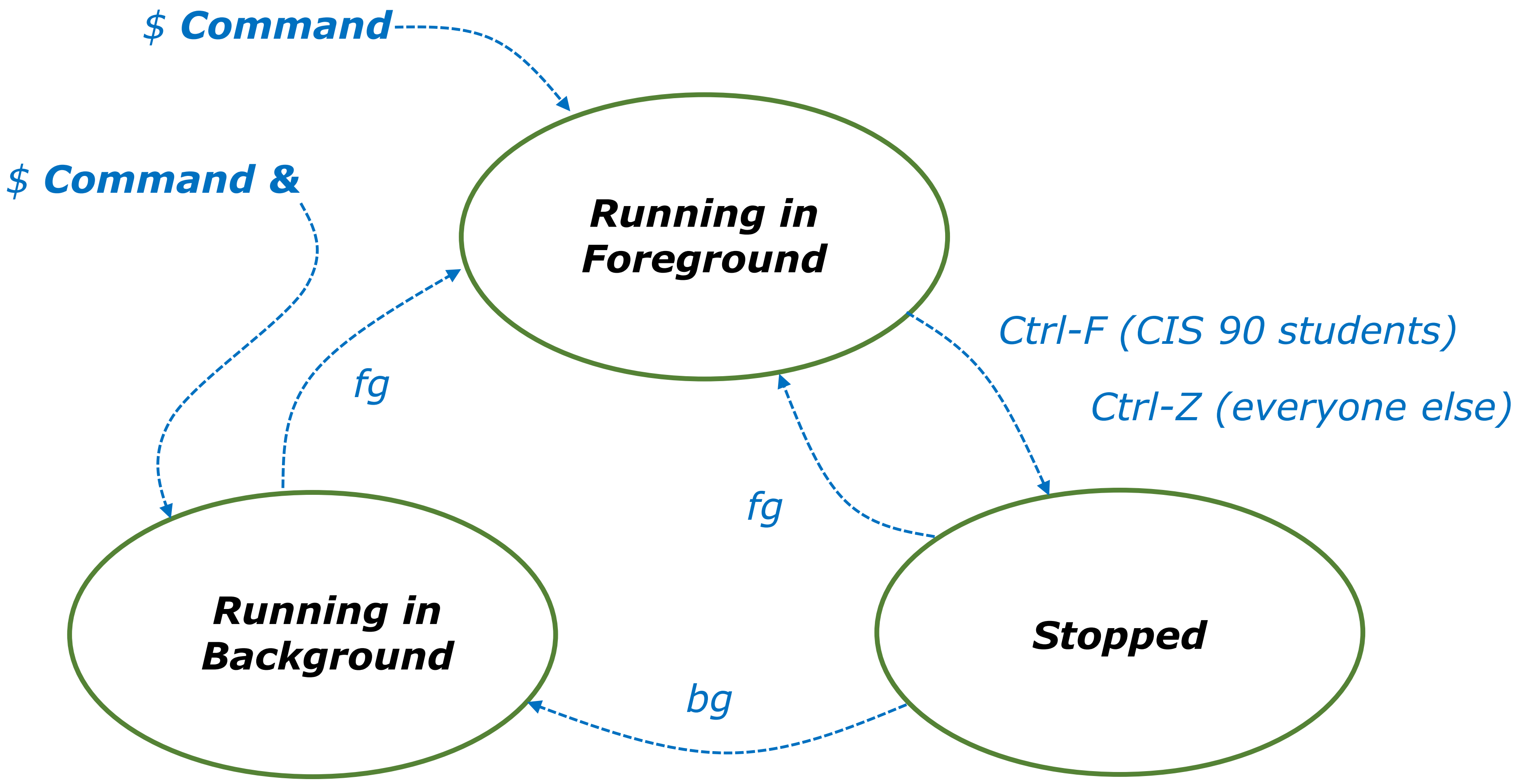
*Use **jobs**, **bg**, **fg** to list and resume jobs in the foreground or background*

Job Control
A feature of the bash shell

*When a process is **running** (status=R) the user can **stop** it (status=T) and choose whether it runs in the **background** or **foreground***



Job Control
A feature of the bash shell



*Use the **jobs** command to view stopped and background jobs*

Job Control

Find out with keystroke combination is configured to suspend a process

```
/home/cis90ol/simmsben $ stty -a
speed 38400 baud; rows 24; columns 80; line = 0;
intr = ^C; quit = ^\; erase = ^?; kill = ^U; eof = ^D; eol = <undef>;
eol2 = <undef>; swtch = <undef>; start = ^Q; stop = ^S; susp = ^F; rprnt = ^R;
werase = ^W; lnext = ^V; flush = ^O; min = 1; time = 0;
-parenb -parodd cs8 -hupcl -cstopb cread -clocal -crtscts -cdtrdsr
-ignbrk -brkint -ignpar -parmrk -inpck -istrip -inlcr -igncr icrnl ixon -ixoff
-iuclc -ixany -imaxbel -iutf8
opost -olcuc -ocrnl onlcr -onocr -onlret -ofill -ofdel nl0 cr0 tab0 bs0 vt0 ff0
isig icanon iexten echo echoe echok -echonl -noflsh -xcase -tostop -echoprt
echoctl echoke
/home/cis90ol/simmsben $
```

In this case it is Ctrl-F that will be used to suspend a process

Put how your suspend keystrokes are configured in the chat window

Job Control
Managing jobs

```
/home/cis90ol/simmsben $ sleep 120
Ctrl-Z or Ctrl-F (to suspend process)
[1]+  Stopped                 sleep 120

/home/cis90ol/simmsben $ sleep 110
Ctrl-Z or Ctrl-F (to suspend process)
[2]+  Stopped                 sleep 110

/home/cis90ol/simmsben $ sleep 100
Ctrl-Z or Ctrl-F (to suspend process)
[3]+  Stopped                 sleep 100

/home/cis90ol/simmsben $ jobs
[1]   Stopped                 sleep 120
[2]-  Stopped                 sleep 110
[3]+  Stopped                 sleep 100
```

Lets start up 3 sleep commands and suspend each of them.

Note: The sleep command is a simple way to run a command that will take awhile to finish.

***sleep 120** will last 120 seconds before it is finished.*

Job Control
Managing jobs

```
/home/cis90ol/simmsben $ jobs
[1]    Stopped                       sleep 120
[2]-   Stopped                       sleep 110
[3]+   Stopped                       sleep 100


/home/cis90ol/simmsben $ ps -l
F S   UID   PID  PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
0 S  1082  5364  5363  0  75   0 -  1168 wait   pts/2    00:00:00 bash
0 T  1082  5452  5364  0  75   0 -   929 finish pts/2    00:00:00 sleep
0 T  1082  5453  5364  0  75   0 -   929 finish pts/2    00:00:00 sleep
0 T  1082  5454  5364  0  75   0 -   929 finish pts/2    00:00:00 sleep
0 R  1082  5459  5364  0  77   0 -  1054 -      pts/2    00:00:00 ps
```

Note, all three processes are sTopped

Job Control
Managing jobs

```
/home/cis90ol/simmsben $ bg 2
[2]- sleep 110 &
/home/cis90ol/simmsben $ jobs
[1]-  Stopped                 sleep 120
[2]   Running                 sleep 110 &
[3]+  Stopped                 sleep 100
```

Let's resume job 2 in the background

```
/home/cis90ol/simmsben $ bg 1
[1]- sleep 120 &
/home/cis90ol/simmsben $ jobs
[1]   Running                 sleep 120 &
[2]-  Running                 sleep 110 &
[3]+  Stopped                 sleep 100
```

Let's resume job 1in the background

```
/home/cis90ol/simmsben $ fg 3
sleep 100
```

Let's resume job 1 in the foreground

At this point we lose control of the keyboard again until sleep 100 is finished

Job Control
Managing jobs

```
/home/cis90ol/simmsben $ jobs
[1]-  Done
sleep 120
[2]+  Done
sleep 110
```

Background jobs are all done!

Load Balancing & Scheduling (Review)

Load Balancing

The **at** command:

- reads from stdin for a list of commands to run
- runs those commands at the specified time
- Any output from those commands will be emailed
- Use **atq** and **atrm** to manage scheduled commands

*Use **at** to schedule commands to run in the future*

Load Balancing
Managing queued jobs

```
at now + 5 minutes

at now + 1 hour

at 7:58AM

at 7:47PM 11/25/2016

at teatime
```

Ways to specify future times

Load Balancing

Managing queued jobs

```
/home/cis90/simben $ atq
25      2011-11-12 14:09 a simben90
28      2011-12-12 03:00 a simben90
27      2011-11-19 12:10 a simben90
26      2011-11-12 16:00 a simben90
24      2011-11-12 12:14 a simben90
```

*The **atq** command lists jobs queued to run in the future*

```
/home/cis90/simben $ atrm 24
/home/cis90/simben $ atq
25      2011-11-12 14:09 a simben90
28      2011-12-12 03:00 a simben90
27      2011-11-19 12:10 a simben90
26      2011-11-12 16:00 a simben90
```

*The **atrm** command is used to remove jobs from the queue*

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

*Note: The **jobs** command lists processes running or suspended in the background and is NOT used for **at** commands.*

Load Balancing

Try it yourself with your own terminal device and username:

These should match

```
[rsimms@oslab ~]$ tty
/dev/pts/xx

[rsimms@oslab ~]$ at now + 2 minutes
at> echo "Take Benji for a walk" | mail -s "walk the dog" $LOGNAME
at> echo "Read your mail" > /dev/pts/xx
at> <EOT>   Ctrl-D
job 11 at 2012-11-05 11:02
[rsimms@oslab ~]$ atq
11      2012-11-05 11:02 a rsimms
[rsimms@oslab ~]$
```

Type what happens in the chat window:

text editors

There are lots of text editors …

Windows
- notepad
- notepad++
- textpad

Mac
- TextWrangler

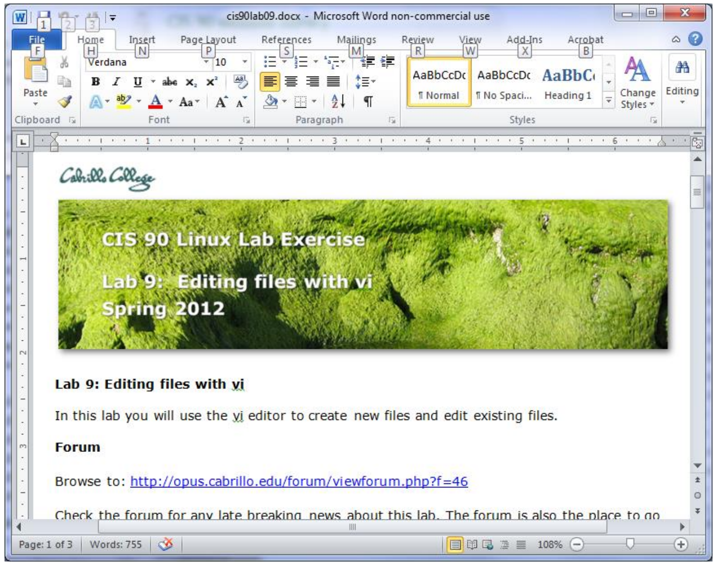
Linux
- gedit
- emacs
- nano
- vi
- jove

Thanks Maria!

Text editors and word processors are different!

- *Word processors are used by many different people to create documents containing text and graphics.*
- *Text editors are used by programmers to develop software and web designers to create web sites.*

CIS 90 - Lesson 11



CIS 90 Linux Lab Exercise

Lab 9: Editing files with vi
Spring 2012

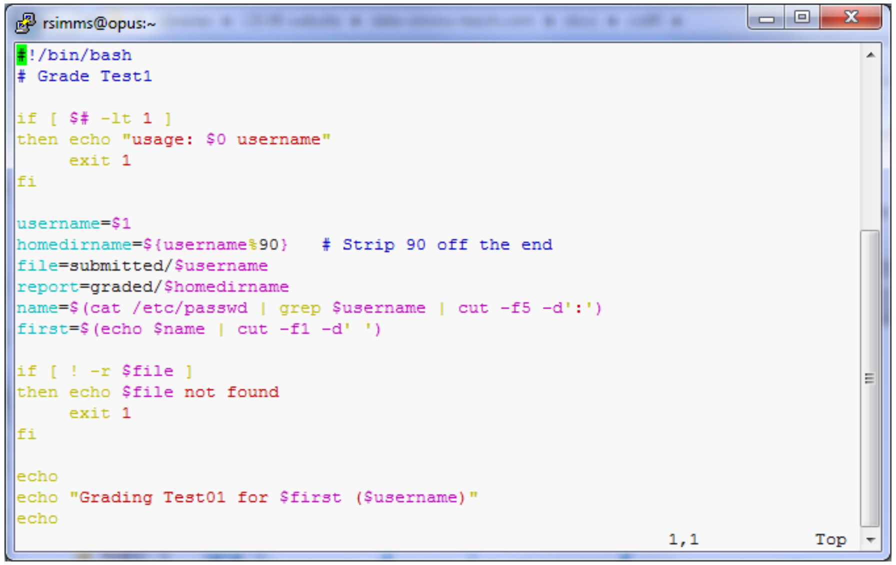
Lab 9: Editing files with vi

In this lab you will use the vi editor to create new files and edit existing files.

Forum

Browse to: http://opus.cabrillo.edu/forum/viewforum.php?f=46

Check the forum for any late breaking news about this lab. The forum is also the place to go



```
#!/bin/bash
# Grade Test1

if [ $# -lt 1 ]
then echo "usage: $0 username"
     exit 1
fi

username=$1
homedirname=${username%90}    # Strip 90 off the end
file=submitted/$username
report=graded/$homedirname
name=$(cat /etc/passwd | grep $username | cut -f5 -d':')
first=$(echo $name | cut -f1 -d' ')

if [ ! -r $file ]
then echo $file not found
     exit 1
fi

echo
echo "Grading Test01 for $first ($username)"
echo
                                                  1,1           Top
```

***Word processors** allow a rich set of formatting (fonts, sizes, styles, color) and graphics to be added to documents.*

***Text editors** use color to show the language syntax*

vi 101

On Opus-II we are actually running VIM

```
/home/cis90/simben $ type -a vi
vi is aliased to `vim'
vi is /bin/vi
/home/cis90/simben $ type vim
vim is hashed (/usr/bin/vim)
```

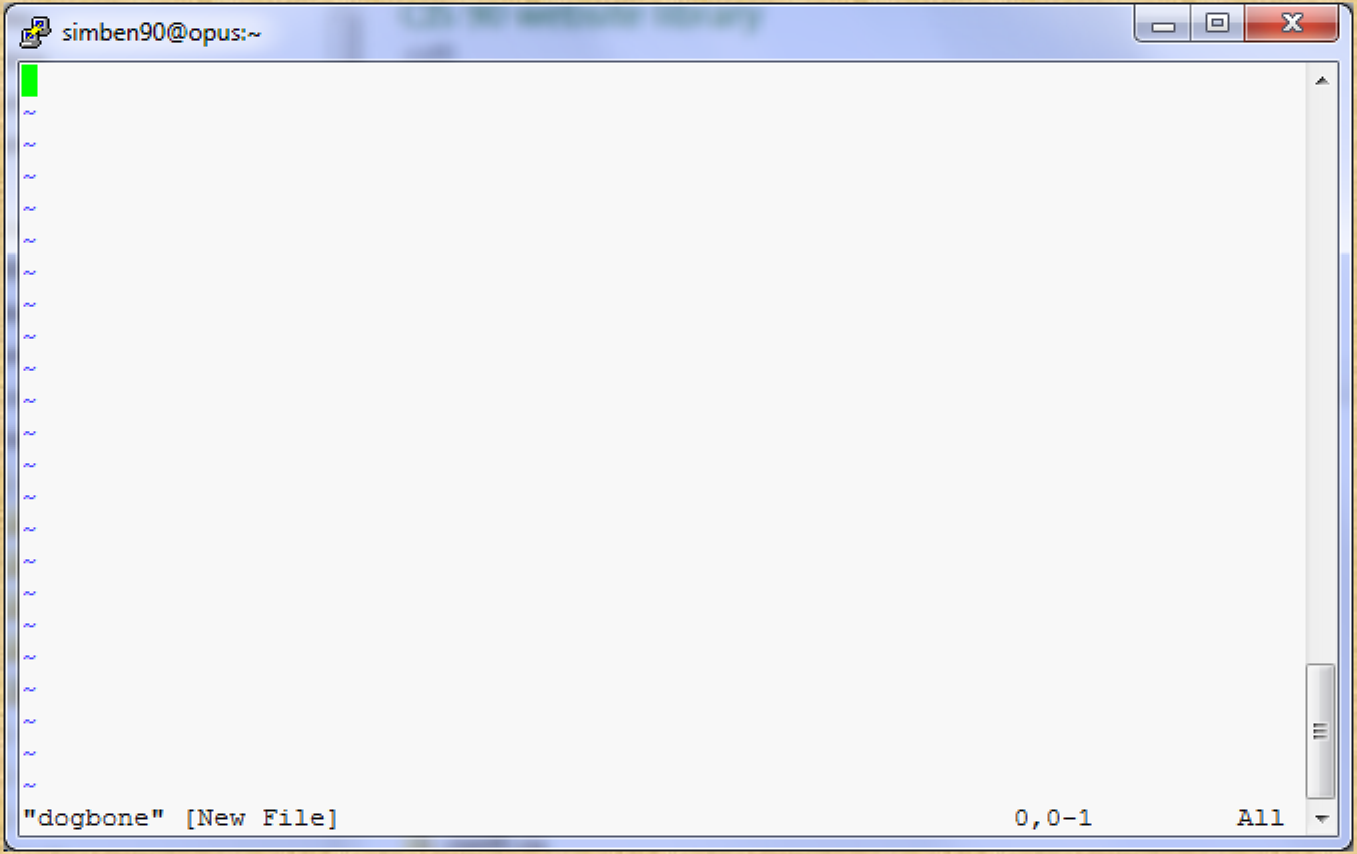
History:
- The original vi code was written by Bill Joy for BSD Unix
- Bill Joy co-founded Sun Microsystems in 1982
- vi (for "visual")
- vim is an enhanced version of vi

```
/home/cis90/simben $ vi dogbone
```

Type this

CIS 90 - Lesson 11

See this …

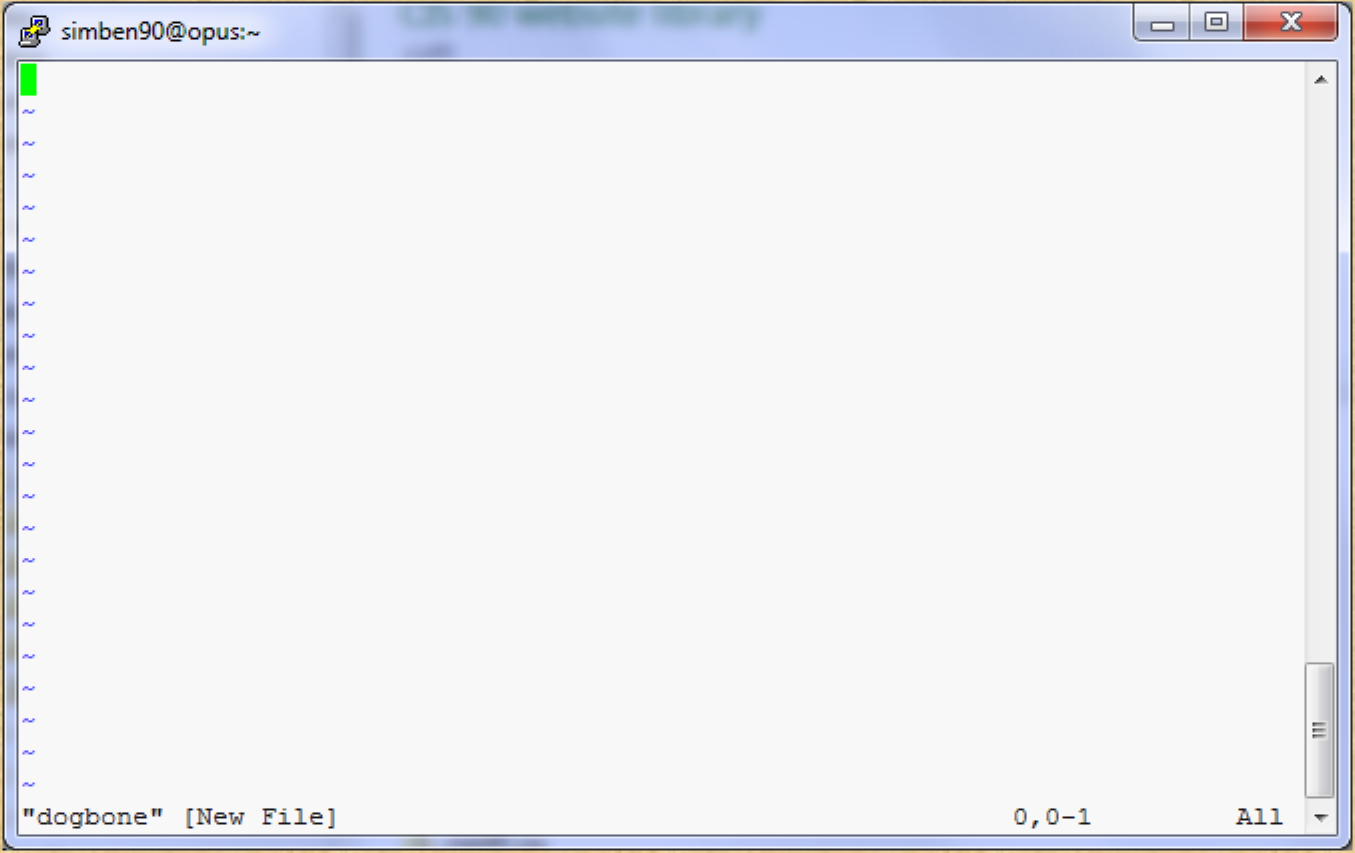


simben90@opus:~

```

~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"dogbone" [New File]                                          0,0-1         All
```

Take your hands OFF THE MOUSE – don't use it in vi!

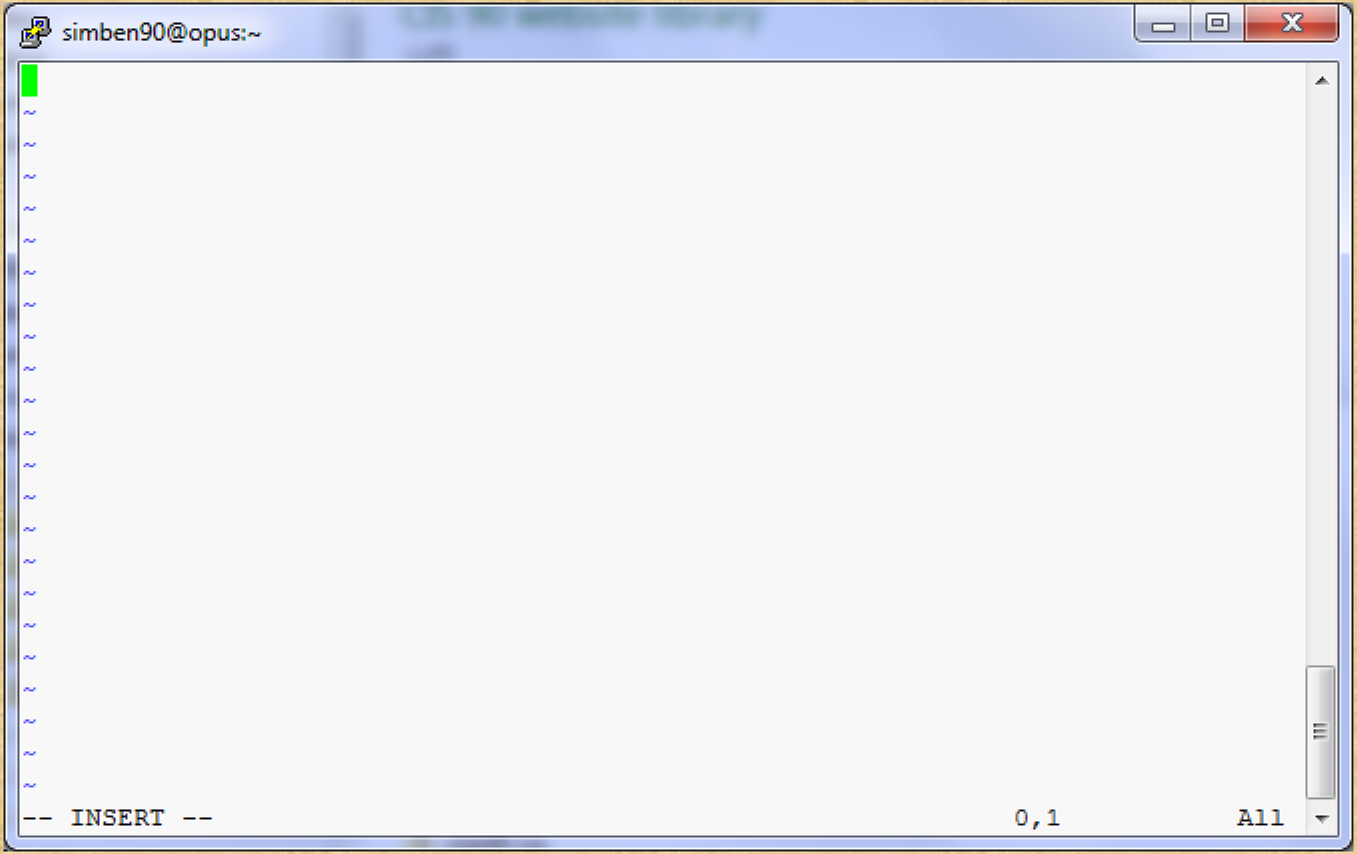
Tap the letter i key (for insert)



```
█
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"dogbone" [New File]                                    0,0-1         All
```

Take your hands OFF THE MOUSE – don't use it in vi!

See this ...

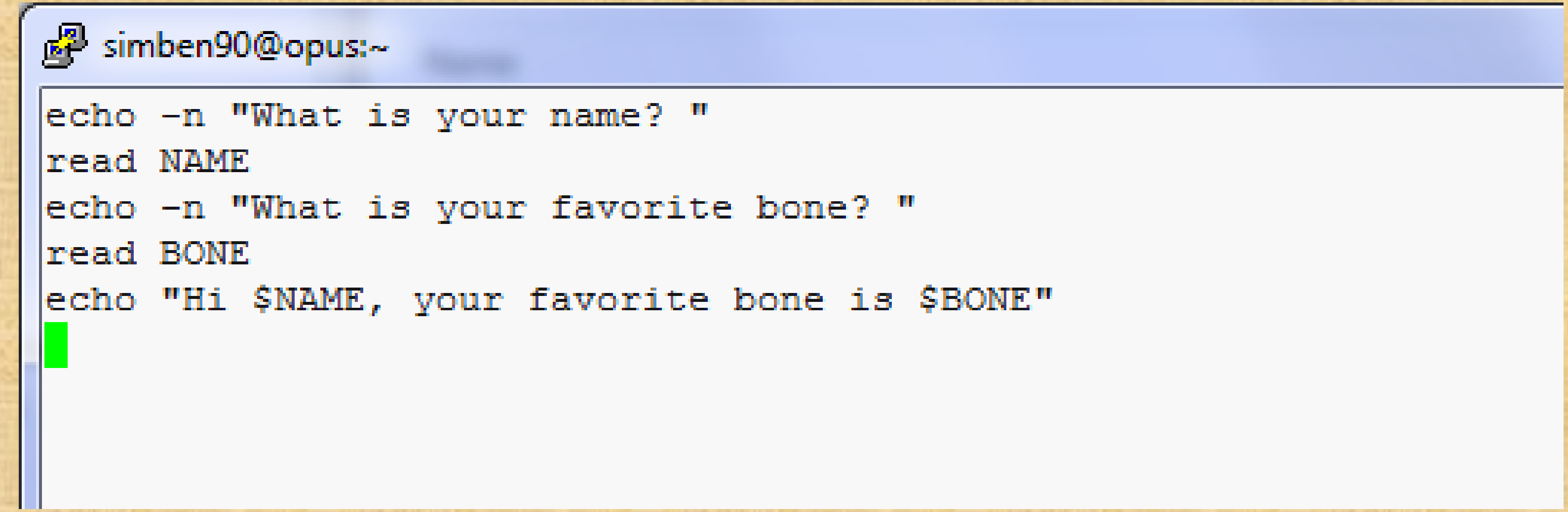


simben90@opus:~

```

~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
-- INSERT --                                                        0,1            All
```

Take your hands OFF THE MOUSE – don't use it in vi!

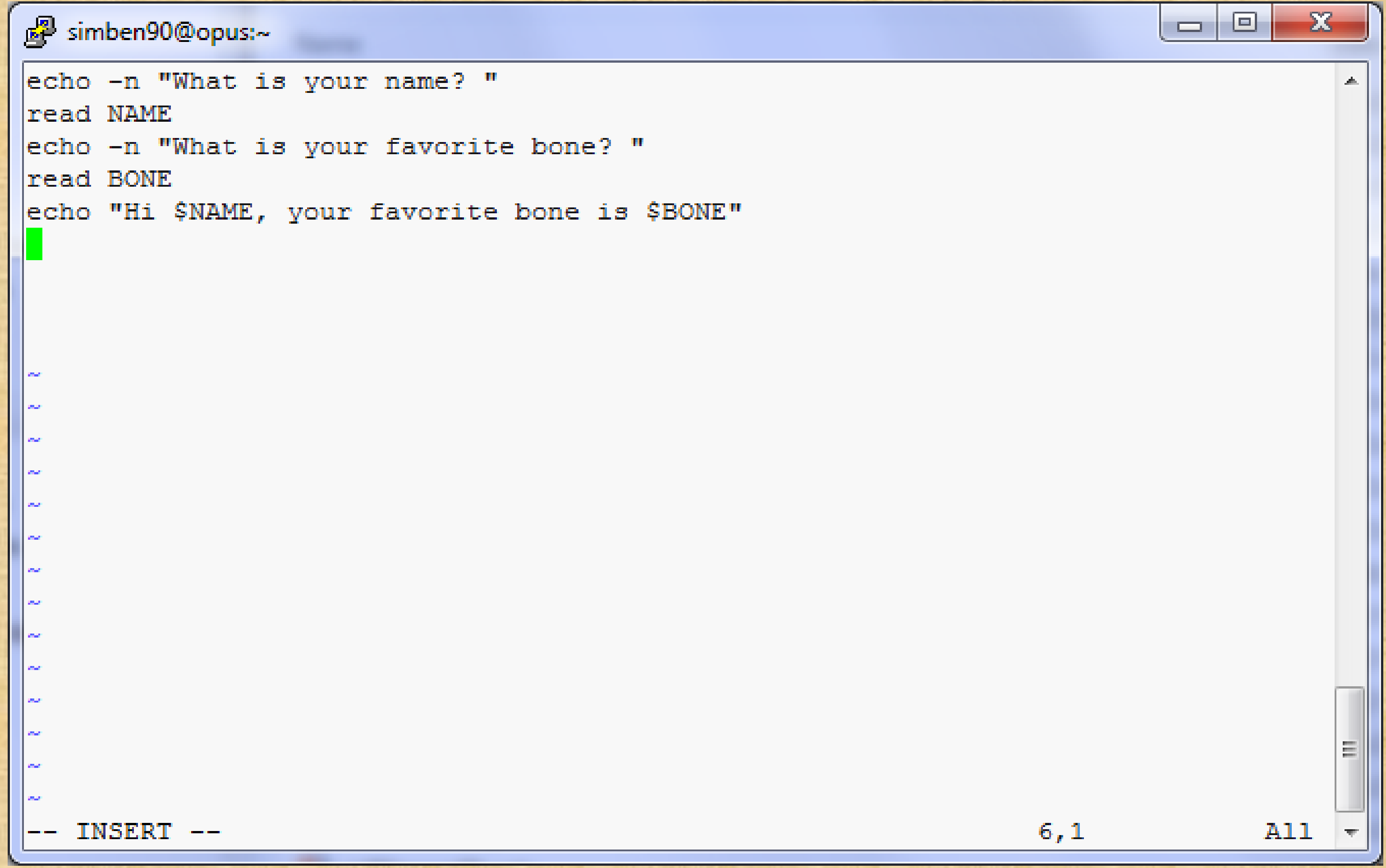
Very carefully type these five lines


simben90@opus:~

```
echo -n "What is your name? "
read NAME
echo -n "What is your favorite bone? "
read BONE
echo "Hi $NAME, your favorite bone is $BONE"
```

Take your hands OFF THE MOUSE – don’t use it in vi!

Have your neighbor check that your five lines are PERFECT

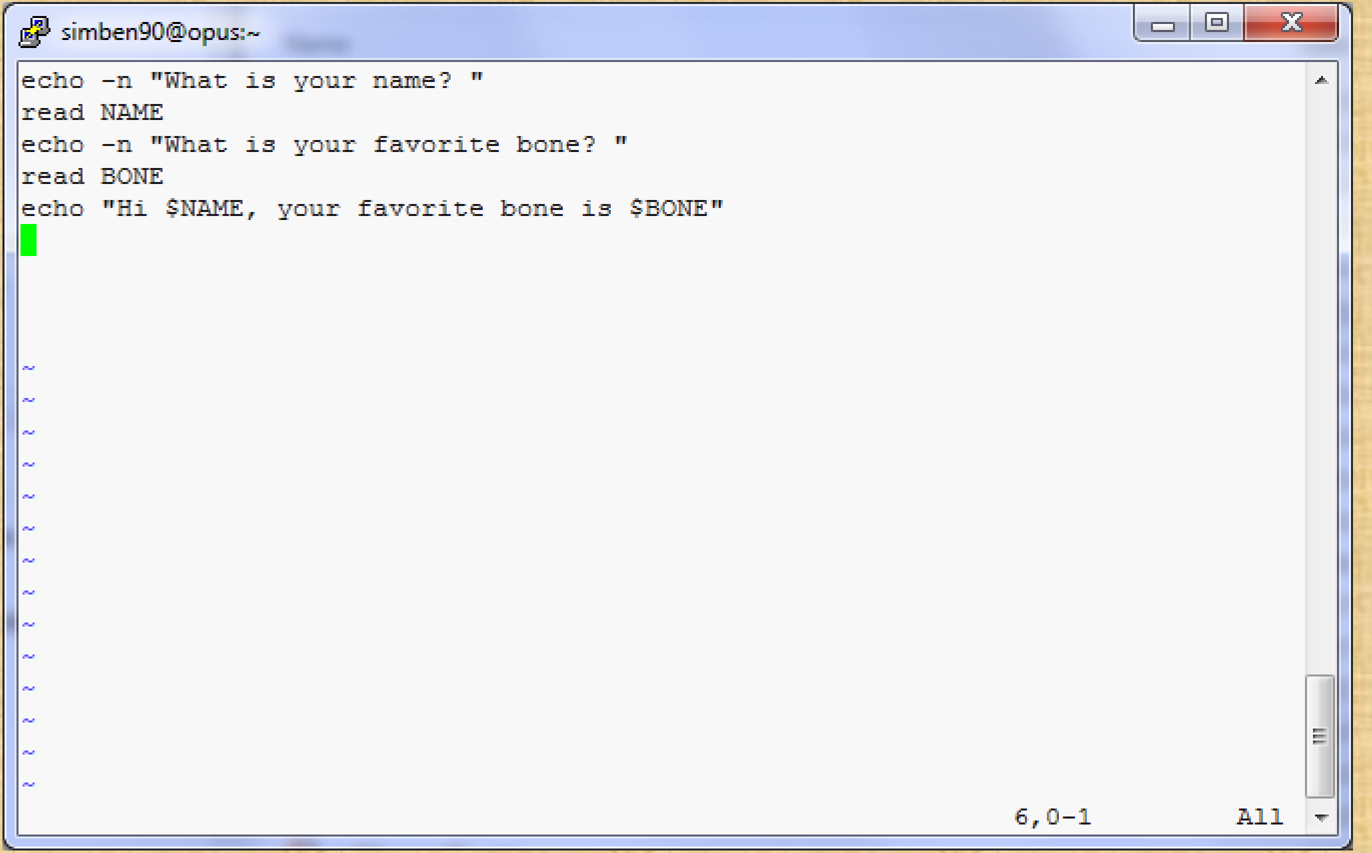


simben90@opus:~

```
echo -n "What is your name? "
read NAME
echo -n "What is your favorite bone? "
read BONE
echo "Hi $NAME, your favorite bone is $BONE"

~
~
~
~
~
~
~
~
~
~
~
~
~
~
-- INSERT --                                       6,1           All
```

Take your hands OFF THE MOUSE – don't use it in vi!

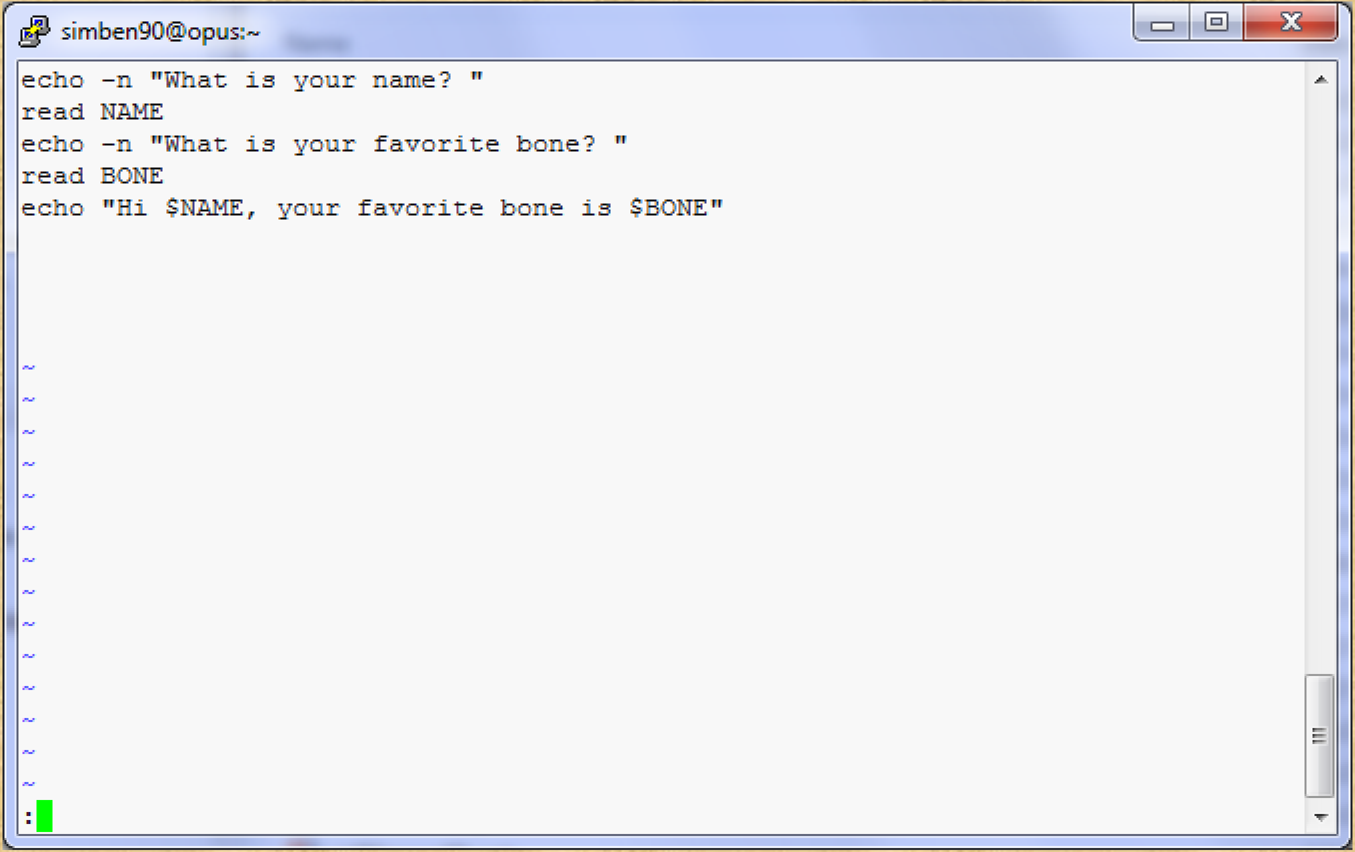
Tap the esc key


simben90@opus:~

```
echo -n "What is your name? "
read NAME
echo -n "What is your favorite bone? "
read BONE
echo "Hi $NAME, your favorite bone is $BONE"

~
~
~
~
~
~
~
~
~
~
~
~
~
~
                                                              6,0-1         All
```

Take your hands OFF THE MOUSE – don't use it in vi!

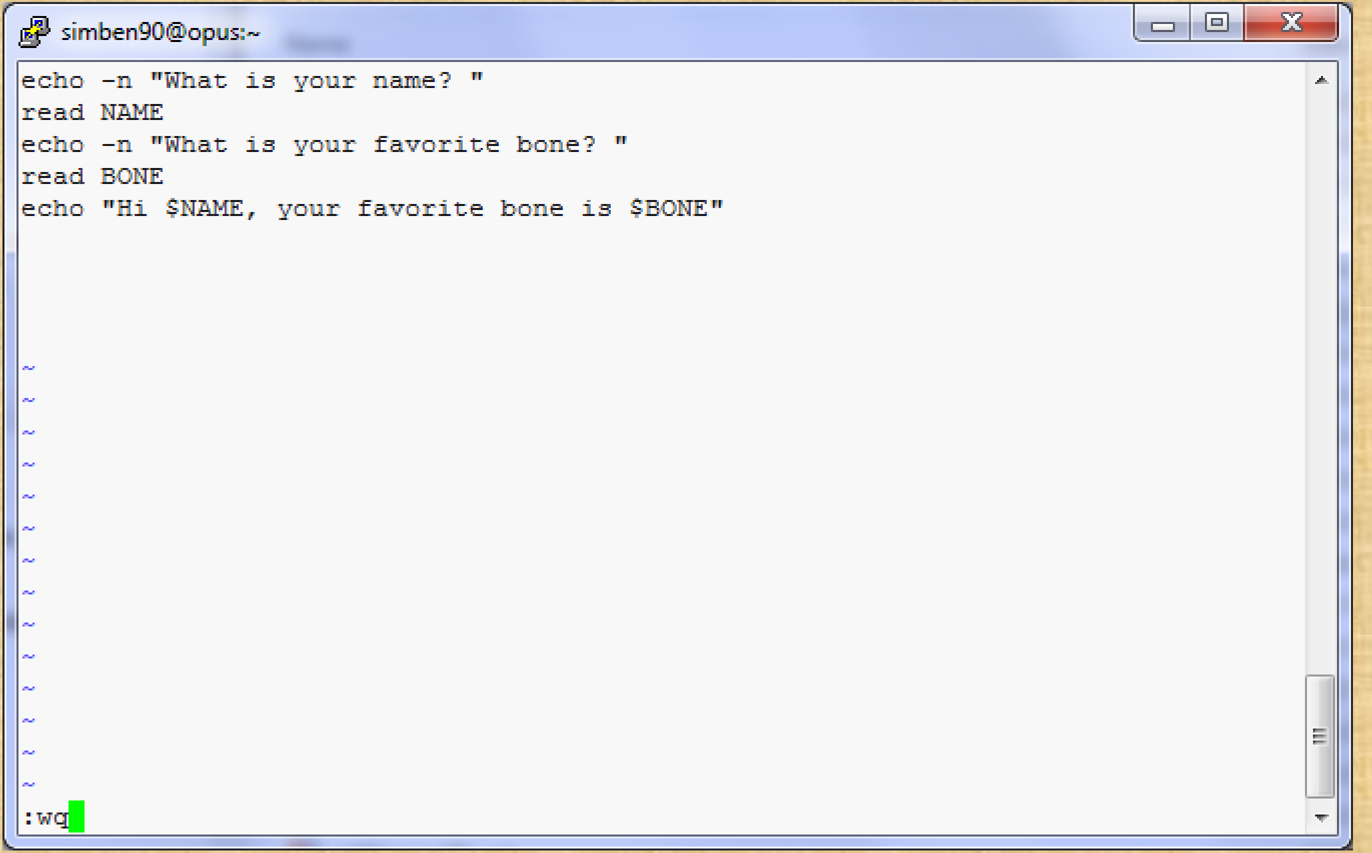
Type a :



```
echo -n "What is your name? "
read NAME
echo -n "What is your favorite bone? "
read BONE
echo "Hi $NAME, your favorite bone is $BONE"







~
~
~
~
~
~
~
~
~
~
~
~
~
~
:
```

Take your hands OFF THE MOUSE – don't use it in vi!

Type wq



```
echo -n "What is your name? "
read NAME
echo -n "What is your favorite bone? "
read BONE
echo "Hi $NAME, your favorite bone is $BONE"





~
~
~
~
~
~
~
~
~
~
~
~
~
~
:wq
```

Take your hands OFF THE MOUSE – don't use it in vi!

Tap the enter key and see …

```
/home/cis90/simben $ vi dogbone
/home/cis90/simben $
```

Add execute permissions and try your new script

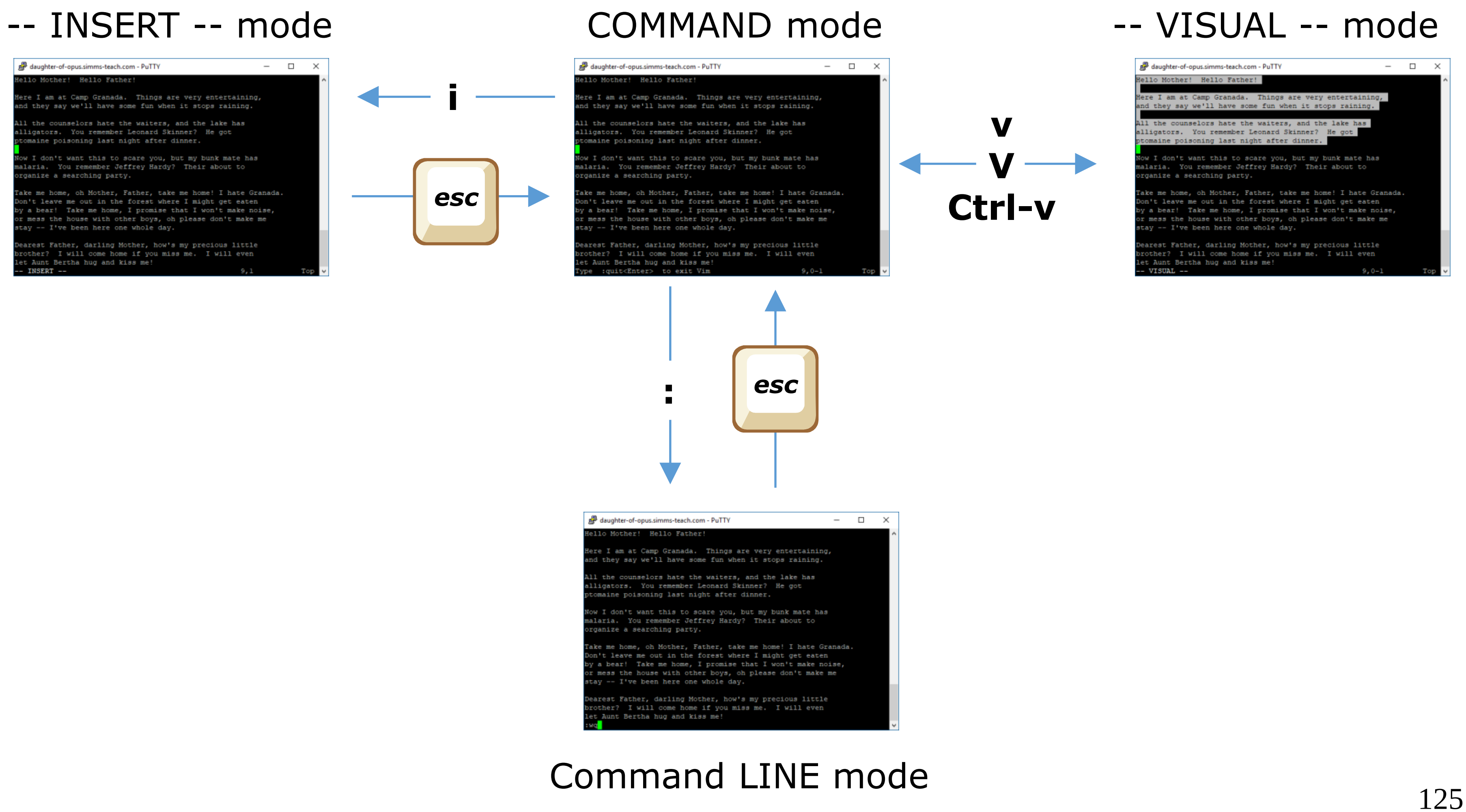
```
/home/cis90/simben $ chmod +x dogbone

/home/cis90/simben $ dogbone
What is your name? Benji
What is your favorite bone? chicken
Hi Benji, your favorite bone is chicken
/home/cis90/simben $
```

vi

COMMAND mode
INSERT mode
command LINE mode

CIS 90 - Lesson 11

```
/home/cis90/simben $ cp letter myletter
/home/cis90/simben $ vi myletter
```



```
Hello Mother!  Hello Father!

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

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

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

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

Dearest Father, darling Mother, how's my precious little
brother?  I will come home if you miss me.  I will even
let Aunt Bertha hug and kiss me!
-- INSERT --                                   9,1           Top
```

```
Type  :quit<Enter>  to exit Vim                 9,0-1         Top
```

```
-- VISUAL --                                    9,0-1         Top
```

```
:wq
```

vi
Moving around in a file

Use in COMMAND mode

h moves the cursor one character to the left
j moves the cursor down one line
k moves the cursor up one line
l moves the cursor one character to the right

^d scrolls down 10 lines
^u scrolls up 10 lines
^f page forward one page
^b page back one page

Try typing a number in front of these commands and notice what happens

With vim (not vi) you can use arrow and page keys instead of these letter commands

Note: ^ is the Ctrl key

vi
Moving around in a file

Use in COMMAND mode

Try typing a number in front of these commands and notice what happens

w moves the cursor one "word" forward
b moves the cursor one "word" back

0 (zero) moves the cursor to the beginning of the line
$ moves the cursor to the end of the line

G moves the cursor to the last line in the file
1G moves the cursor to the first line in the file
105G moves the cursor to line 105

vi
Saving and Quitting

Use in command LINE mode

:w writes any changes to the file you are editing (like Save)

:q quits vi if you have saved your changes
:q! quits vi even if you haven’t saved changes

:wq writes and quits
:wq! writes and quits vi even if you haven’t saved changes

vi
Reading in and Writing out files

Use in command LINE mode

:w *filename* saves your file to a new name (like Save As)
:w! *filename* saves your file to a new name overwriting any previous data

:r *filename* reads in the contents of *filename* starting from the cursor position

:e *filename* replaces the current content with the content from *filename*

:%s /*string1*/*string2*/g replaces all string1 with string2 in the file

vi
Entering INSERT mode

From COMMAND mode.

i Ready to insert characters immediately before the current cursor position
I Ready to insert characters at the start of the current line

a Ready to append characters immediately after the current cursor position
A Ready to append characters at the end of the current line

o Ready to input characters in a new line that opens up below the cursor
O Ready to input characters in a new line that opens up above the cursor

vi
Cut, Copy, Pasting Commands

Use in COMMAND mode

x Deletes the current character
r Replace the current character with the character you type next

dw Deletes the current word
dd Deletes the current line

D Deletes to the end of the line

yy Copies a line to the clipboard buffer
p Pastes whatever is in the clipboard buffer below the current cursor
P Pastes whatever is in the clipboard buffer above the current cursor

vi
Miscellaneous Useful Commands

Use in COMMAND mode.

^g Tells you the filename you are editing and what line your cursor is on

u Undoes the last command you executed
^r Undo the undo (redo)

. Repeats the last command you executed

/string Searches for the string of characters in the file
n Finds the next occurrence of the current search string looking down the file
N Finds the next occurrence of the current search string looking up the file

~ Changes the case of the current character

Note: ^ is the Ctrl key

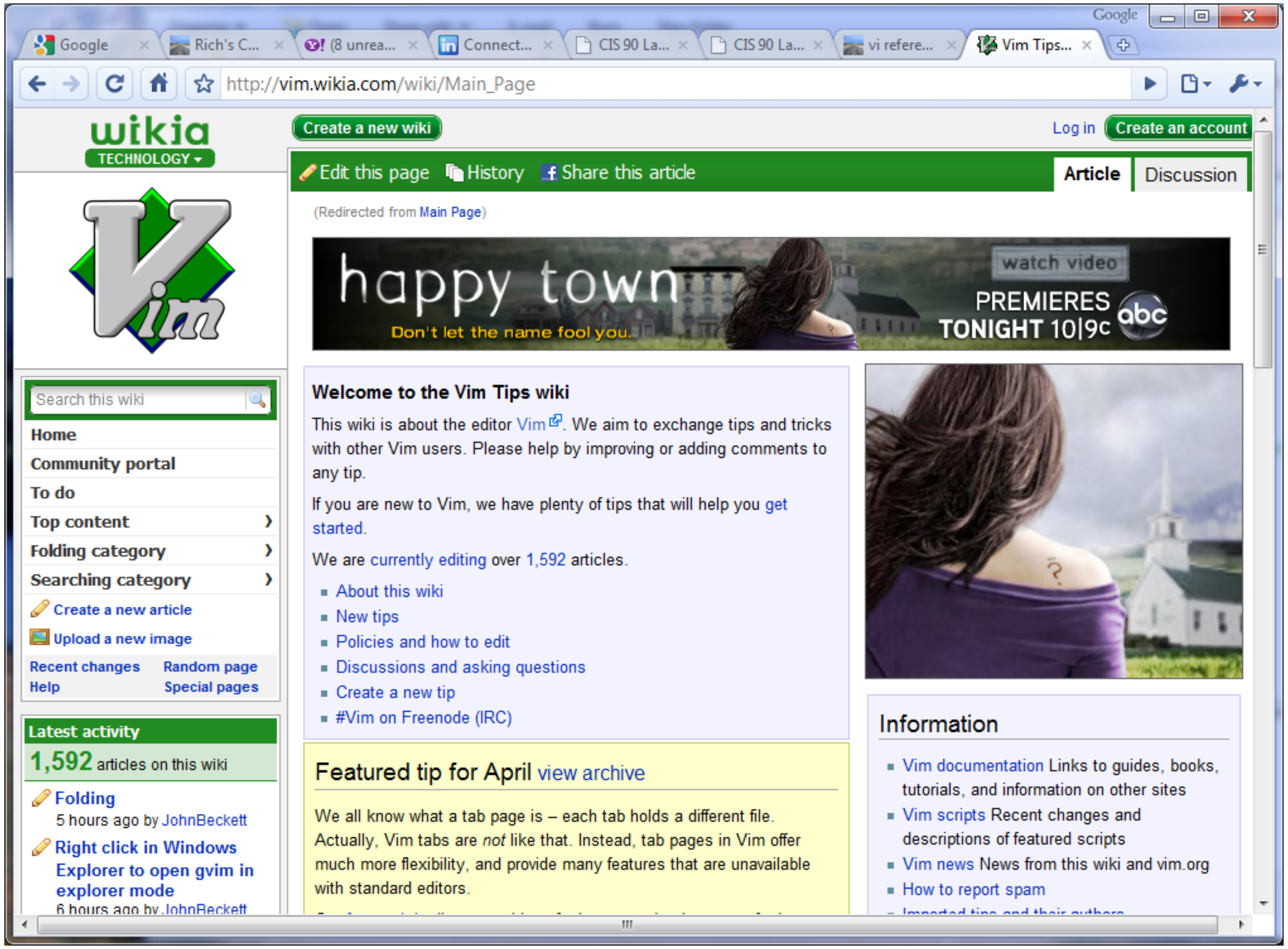
Use vi to edit your *edits/text.err* file

```
This is line number1.
This is line number 1.
Thi sis line line number 2.
his is line number3.line number3.
This is This is line #4.
this number5 is line .
Here is line number     6.
This is lamw number     7.
Thi is line nunber9.
This is line
number10.
```

```
This is line number 1.
This is line number 2.
This is line number 3.
This is line number 4.
This is line number 5.
This is line number 6.
This is line number 7.
This is line number 8.
This is line number 9.
This is line number 10.
```

Copy your corrected file into the chat window when finished

http://vim.wikia.com/wiki/Main_Page



(Redirected from Main Page)

Welcome to the Vim Tips wiki

This wiki is about the editor Vim. We aim to exchange tips and tricks with other Vim users. Please help by improving or adding comments to any tip.

If you are new to Vim, we have plenty of tips that will help you get started.

We are currently editing over 1,592 articles.

- About this wiki
- New tips
- Policies and how to edit
- Discussions and asking questions
- Create a new tip
- #Vim on Freenode (IRC)

Featured tip for April view archive

We all know what a tab page is – each tab holds a different file. Actually, Vim tabs are *not* like that. Instead, tab pages in Vim offer much more flexibility, and provide many features that are unavailable with standard editors.

Latest activity

1,592 articles on this wiki

Folding
5 hours ago by JohnBeckett

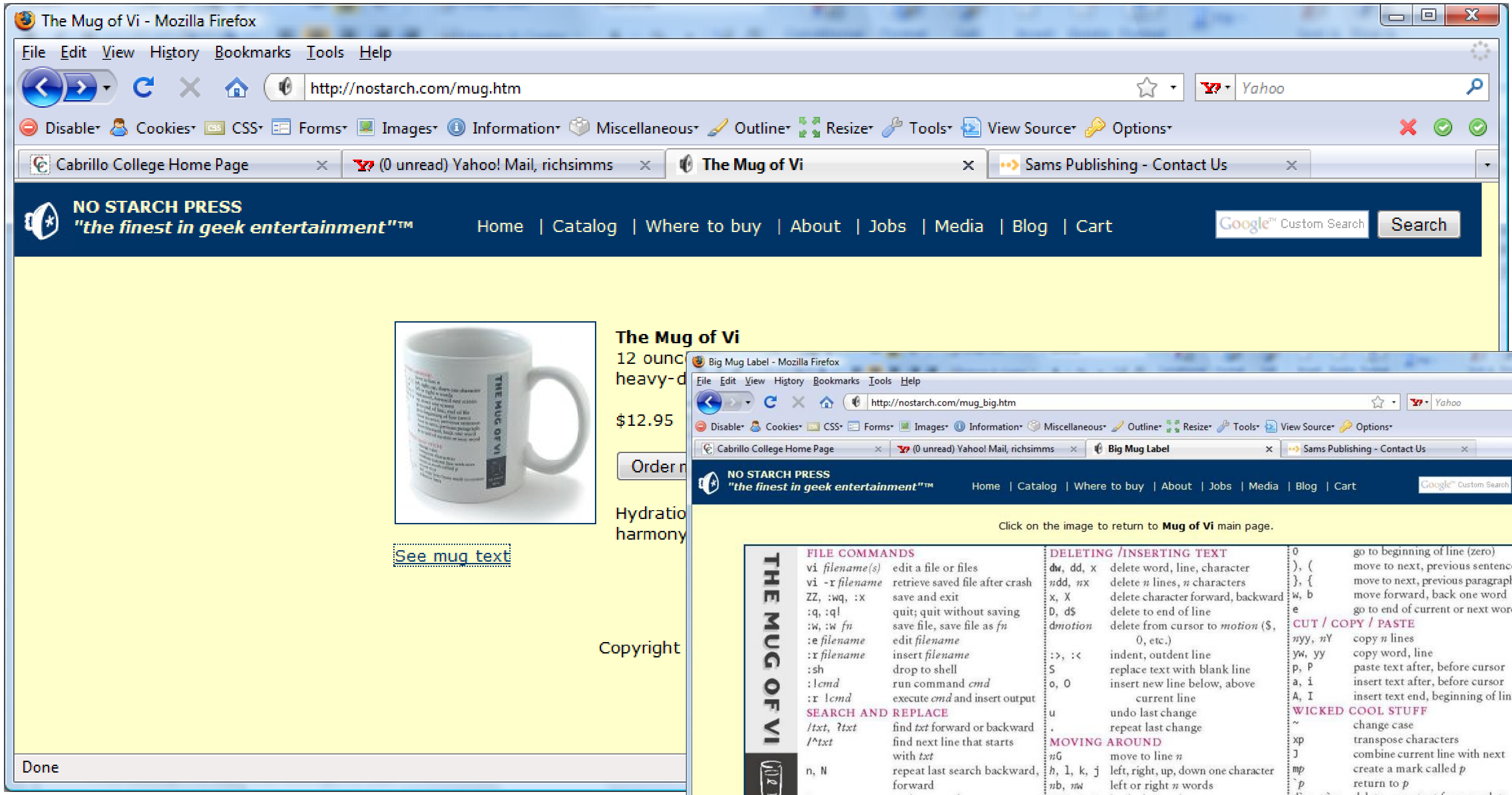
Right click in Windows Explorer to open gvim in explorer mode
6 hours ago by JohnBeckett

Information

- Vim documentation Links to guides, books, tutorials, and information on other sites
- Vim scripts Recent changes and descriptions of featured scripts
- Vim news News from this wiki and vim.org
- How to report spam

Tips and tricks for VIM users

The Mug of vi


The Mug of Vi - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://nostarch.com/mug.htm
Cabrillo College Home Page
(0 unread) Yahoo! Mail, richsimms
The Mug of Vi
Sams Publishing - Contact Us
NO STARCH PRESS
"the finest in geek entertainment"™
Home | Catalog | Where to buy | About | Jobs | Media | Blog | Cart
Search
Done

The Mug of Vi
12 ounc
heavy-d

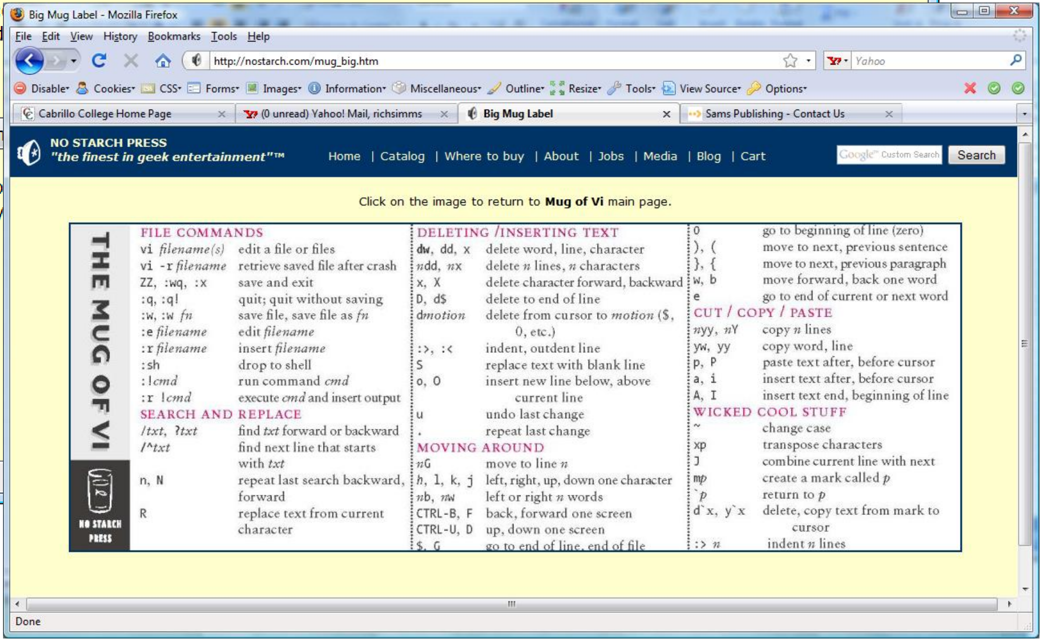
$12.95

Order n

Hydratio
harmony

See mug text

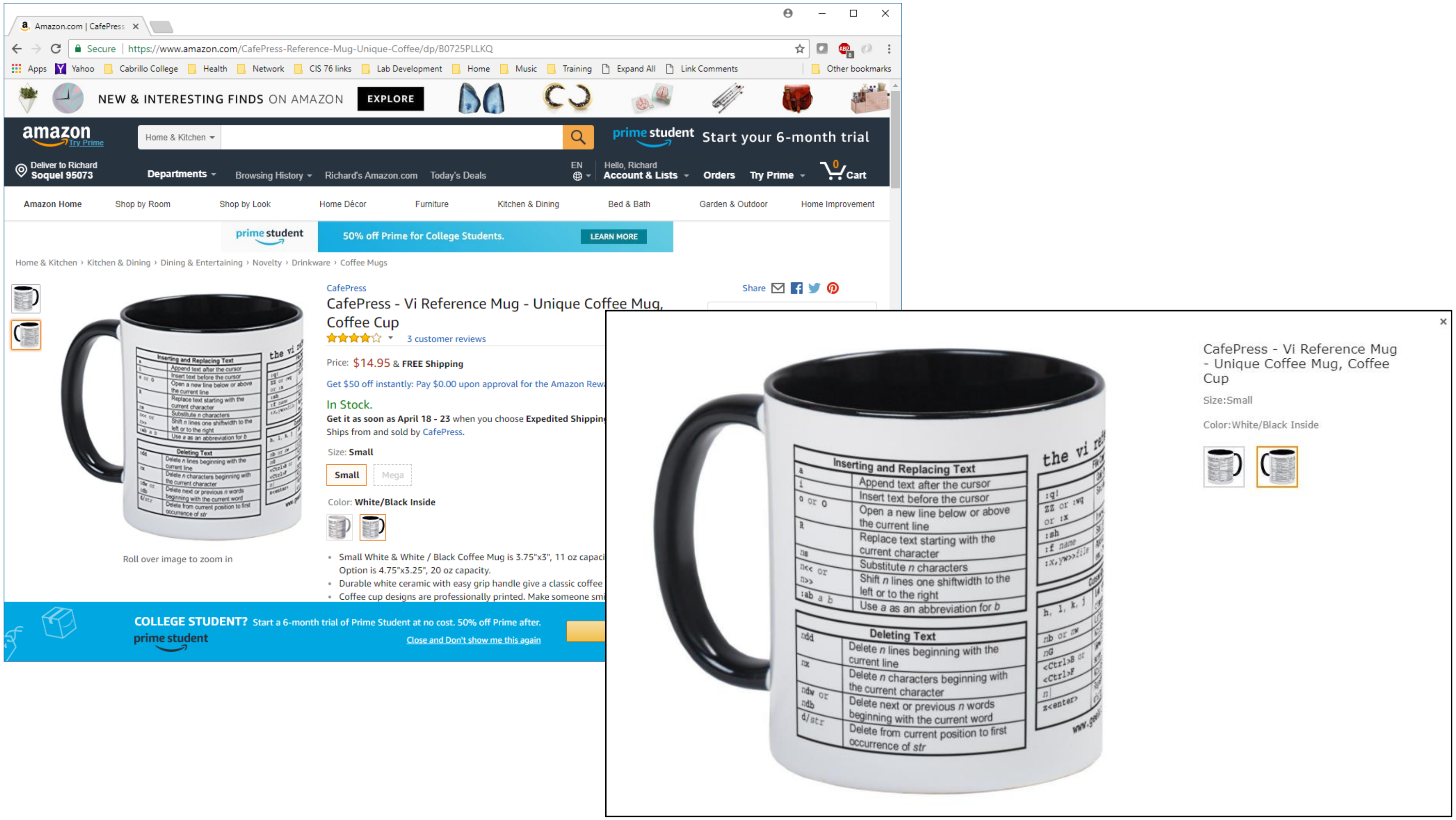
Copyright


Big Mug Label - Mozilla Firefox
http://nostarch.com/mug_big.htm
Big Mug Label
NO STARCH PRESS
"the finest in geek entertainment"™
Home | Catalog | Where to buy | About | Jobs | Media | Blog | Cart
Search
THE MUG OF VI
NO STARCH PRESS
Done

Click on the image to return to **Mug of Vi** main page.

FILE COMMANDS	
`vi` *filename(s)*	edit a file or files
`vi -r` *filename*	retrieve saved file after crash
`ZZ, :wq, :x`	save and exit
`:q, :q!`	quit; quit without saving
`:w, :w` *fn*	save file, save file as *fn*
`:e` *filename*	edit *filename*
`:r` *filename*	insert *filename*
`:sh`	drop to shell
`:!`*cmd*	run command *cmd*
`:r !`*cmd*	execute *cmd* and insert output
SEARCH AND REPLACE	
/*txt*, ?*txt*	find *txt* forward or backward
/^*txt*	find next line that starts with *txt*
`n, N`	repeat last search backward, forward
`R`	replace text from current character
DELETING /INSERTING TEXT	
`dw, dd, x`	delete word, line, character
n`dd`, *n*`x`	delete *n* lines, *n* characters
`x, X`	delete character forward, backward
`D, d$`	delete to end of line
`d`*motion*	delete from cursor to *motion* ($, 0, etc.)
`:>, :<`	indent, outdent line
`S`	replace text with blank line
`o, O`	insert new line below, above current line
`u`	undo last change
`.`	repeat last change
MOVING AROUND	
n`G`	move to line *n*
`h, l, k, j`	left, right, up, down one character
n`b`, *n*`w`	left or right *n* words
`CTRL-B, F`	back, forward one screen
`CTRL-U, D`	up, down one screen
`$, G`	go to end of line, end of file
`0`	go to beginning of line (zero)
`), (`	move to next, previous sentence
`}, {`	move to next, previous paragraph
`w, b`	move forward, back one word
`e`	go to end of current or next word
CUT / COPY / PASTE	
n`yy`, *n*`Y`	copy *n* lines
`yw, yy`	copy word, line
`p, P`	paste text after, before cursor
`a, i`	insert text after, before cursor
`A, I`	insert text end, beginning of line
WICKED COOL STUFF	
`~`	change case
`xp`	transpose characters
`J`	combine current line with next
`m`*p*	create a mark called *p*
`` ` ``*p*	return to *p*
``d`x, y`x``	delete, copy text from mark to cursor
`:>` *n*	indent *n* lines

CafePress - VI Reference Mug



https://www.amazon.com/CafePress-Reference-Mug-Unique-Coffee/dp/B0725PLLKQ

Home & Kitchen › Kitchen & Dining › Dining & Entertaining › Novelty › Drinkware › Coffee Mugs

CafePress

CafePress - Vi Reference Mug - Unique Coffee Mug, Coffee Cup

3 customer reviews

Price: $14.95 & **FREE Shipping**

Get $50 off instantly: Pay $0.00 upon approval for the Amazon Rew

In Stock.

Get it as soon as April 18 - 23 when you choose **Expedited Shippin**

Ships from and sold by CafePress.

Size: **Small**

Color: **White/Black Inside**

- Small White & White / Black Coffee Mug is 3.75"x3", 11 oz capaci Option is 4.75"x3.25", 20 oz capacity.
- Durable white ceramic with easy grip handle give a classic coffee
- Coffee cup designs are professionally printed. Make someone smi

Roll over image to zoom in

CafePress - Vi Reference Mug - Unique Coffee Mug, Coffee Cup

Size:Small

Color:White/Black Inside

Amazon.com | CafePress
Secure
Apps
Yahoo
Cabrillo College
Health
Network
CIS 76 links
Lab Development
Home
Music
Training
Expand All
Link Comments
Other bookmarks
NEW & INTERESTING FINDS ON AMAZON
EXPLORE
amazon
Try Prime
Home & Kitchen
prime student
Start your 6-month trial
Deliver to Richard
Soquel 95073
Departments
Browsing History
Richard's Amazon.com
Today's Deals
EN
Hello, Richard
Account & Lists
Orders
Try Prime
Cart
Amazon Home
Shop by Room
Shop by Look
Home Décor
Furniture
Kitchen & Dining
Bed & Bath
Garden & Outdoor
Home Improvement
50% off Prime for College Students.
LEARN MORE
Share
Small
Mega
COLLEGE STUDENT? Start a 6-month trial of Prime Student at no cost. 50% off Prime after.
Close and Don't show me this again
Inserting and Replacing Text
Append text after the cursor
Insert text before the cursor
Open a new line below or above the current line
Replace text starting with the current character
Substitute n characters
Shift n lines one shiftwidth to the left or to the right
Use a as an abbreviation for b
Deleting Text
Delete n lines beginning with the current line
Delete n characters beginning with the current character
Delete next or previous n words beginning with the current word
Delete from current position to first occurrence of str
the vi ref

/bin/mail and vi

```
/home/cis90/simben $ mail milhom90
Subject: Good Bones
Hey Homer,
I really appreciate thatbone you sent me last week.
Let me knwo if you want to go mark some fench posts
this weekend.
Later,
Ben
```

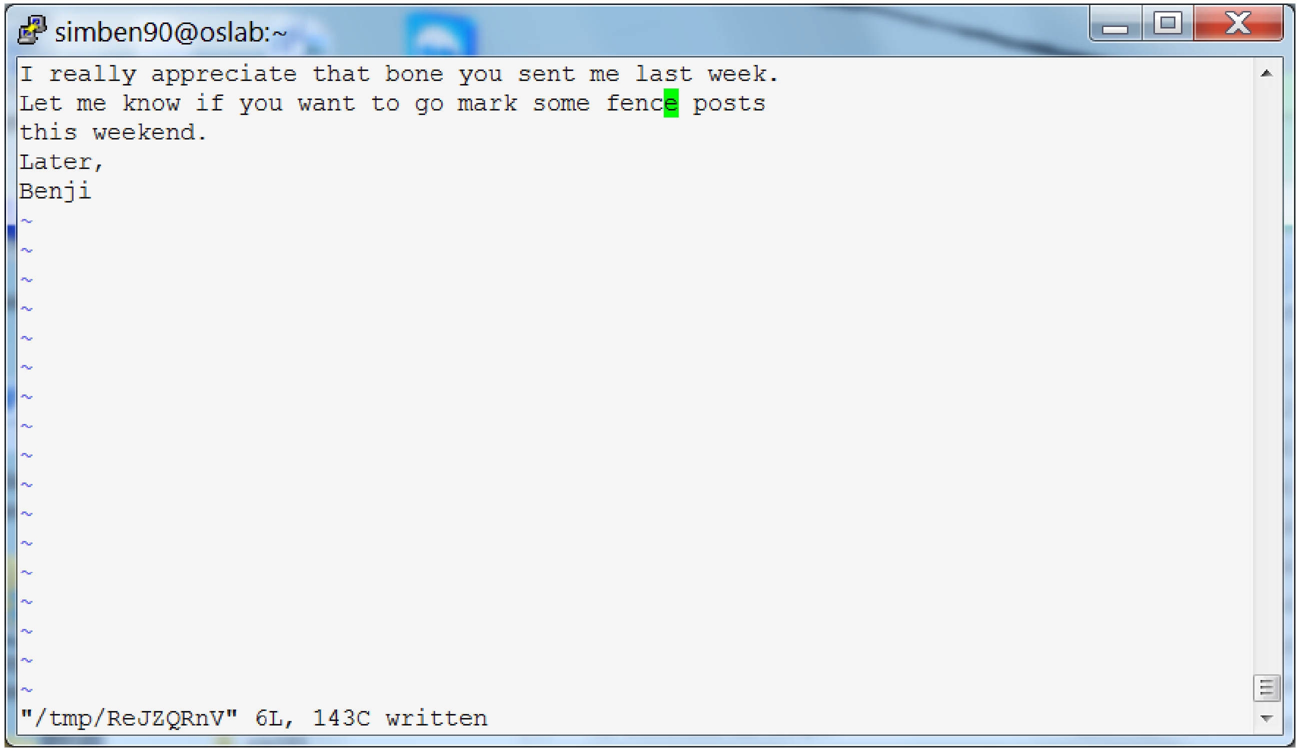
You are composing a message and you spot some typos …
CRUD … what can you do?

/bin/mail and vi

```
/home/cis90/simben $ mail milhom90
Subject: Good Bones
Hey Homer,
I really appreciate thatbone you sent me last week.
Let me knwo if you want to go mark some fench posts
this weekend.
Later,
Ben
~v
```

Well … you could try the ~v command

/bin/mail and vi



```
I really appreciate that bone you sent me last week.
Let me know if you want to go mark some fence posts
this weekend.
Later,
Benji
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"/tmp/ReJZQRnV" 6L, 143C written
```

The message is loaded into vi where changes or additions can be made. :wq is used to save and quit vi

/bin/mail and vi

```
/home/cis90/simben $ mail milhom90
Subject: Good Bones
Hey Homer,
I really appreciate thatbone you sent me last week.
Let me knwo if you want to go mark some fench posts
this weekend.
Later,
Ben
~v
(continue)
.
EOT
/home/cis90/simben $
```

The earlier text with typos is still showing, however the corrected version is what is actually sent.

/bin/mail and vi

```
/home/cis90/milhom $ mail
Heirloom Mail version 12.4 7/29/08.  Type ? for help.
"/var/spool/mail/milhom90": 157 messages 5 new 155 unread
>N157 Benji Simms           Mon Nov 10 14:05  25/952   "Good Bones"
& 157
Message 157:
From simben90@oslab.cis.cabrillo.edu  Mon Nov 10 14:05:20 2014
Return-Path: <simben90@oslab.cis.cabrillo.edu>
From: Benji Simms <simben90@oslab.cis.cabrillo.edu>
Date: Mon, 10 Nov 2014 14:05:20 -0800
To: milhom90@oslab.cis.cabrillo.edu
Subject: Good Bones
User-Agent: Heirloom mailx 12.4 7/29/08
Content-Type: text/plain; charset=us-ascii
Status: R

Hey Homer,
I really appreciate that bone you sent me last week.
Let me know if you want to go mark some fence posts
this weekend.
Later,
Benji

&
```

The message Homer reads has all the typos fixed.

Fix an email message before sending

```
/home/cis90/simben/edits $ mail rsimms
Subject: test of vi
sdkfjas;dflkjas;lkdfj
~v
(continue)
.
EOT
/home/cis90/simben/edits $
```

Once in vi:

- ***Use i to enter insert mode***
- ***make changes***
- ***save with <Esc>:wq***

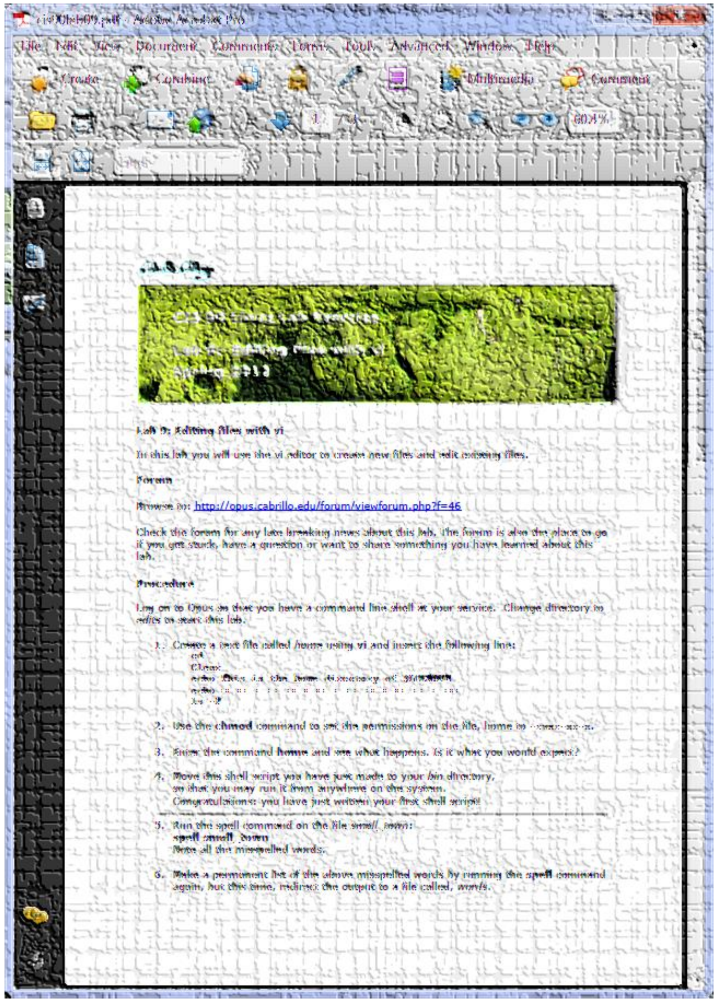
Assignment

Instructor: remember to mail students the tech file!

~/cis90/lab09/mail-langs-all

or

```
at <end-of-class>
at> /home/rsimms/cis90/lab09/mail-langs-all
at> <Ctrl-D>
```



Lab 9: Editing files with vi

In this lab you will use the vi editor to create new files and edit existing files.

Forum

Browse to: http://opus.cabrillo.edu/forum/viewforum.php?f=46

Check the forum for any late breaking news about this lab. The forum is also the place to go if you get stuck, have a question or want to share something you have learned about this lab.

Procedure

Log on to Opus so that you have a command line shell at your service. Change directory to *edits* to start this lab.

1. Create a text file called *home* using **vi** and insert the following lines:
   ```
   cd
   Clear
   echo This is the home directory of [illegible]
   echo [illegible]
   ls -F
   ```
2. Use the **chmod** command to set the permissions on the file, home to [illegible].
3. Enter the command **home** and see what happens. Is it what you would expect?
4. Move this shell script you have just made to your *bin* directory, so that you may run it from anywhere on the system. Congratulations: you have just written your first shell script!

---

5. Run the spell command on the file *small_town*:
   spell small_town
   Note all the misspelled words.
6. Make a permanent list of the above misspelled words by running the **spell** command again, but this time, redirect the output to a file called, *words*.

Lab 9 will help you start building your vi skills!

Wrap up

New commands:	
vi	Run vi editor
New Files and Directories:	
na	na

Next Class

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

Lab 9
Five Posts

Quiz questions for next class:

- How do you send a SIGKILL signal to one of your own processes?
- What vi command is used to exit vi without saving any of the changes you made?
- What vi commands are used for copy and paste?

Backup

The **mystery** of Ctrl-Z vs Ctrl-F

Signals
Special keystrokes

Note: ^ is the Ctrl key

```
/home/cis90/roddyduk $ stty -a
speed 38400 baud; rows 26; columns 78; line = 0;
intr = ^C; quit = ^\; erase = ^?; kill = ^U; eof = ^D; eol = <undef>;
eol2 = <undef>; swtch = <undef>; start = ^Q; stop = ^S; susp = ^F; rprnt = ^R;
werase = ^W; lnext = ^V; flush = ^O; min = 1; time = 0;
```

Ctrl-f

```
[rsimms@opus ~]$ stty -a
speed 38400 baud; rows 39; columns 84; line = 0;
intr = ^C; quit = ^\; erase = ^?; kill = ^U; eof = ^D; eol = <undef>; eol2 = <undef>;
swtch = <undef>; start = ^Q; stop = ^S; susp = ^Z; rprnt = ^R; werase = ^W;
lnext = ^V; flush = ^O; min = 1; time = 0;
```

Ctrl-z

Why does the keystroke to send a Suspend (SIGTSTP or 20) signal differ between roddyduk (Ctrl-F) and rsimms (Ctrl-Z)?

Job Control
A feature of the bash shell

Ctrl-Z or Ctrl-F (sends SIGTSTP 20 signal)
- Stops (suspends) a foreground process

```
[rsimms@opus ~]$ sleep 5

[1]+  Stopped                 sleep 5
```

Ctrl-Z is tapped which stops the sleep command

PID 7728 is stopped

```
[rsimms@opus ~]$ ps -l  -u rsimms
F S   UID   PID  PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
5 S   201  5368  5365  0  75   0 -  2460 -      ?        00:00:00 sshd
0 S   201  5369  5368  0  76   0 -  1165 wait   pts/0    00:00:00 bash
5 S   201  6203  6200  0  75   0 -  2491 -      ?        00:00:00 sshd
0 S   201  6204  6203  0  75   0 -  1165 -      pts/6    00:00:00 bash
0 T   201  7728  6204  0  75   0 -   926 finish pts/6    00:00:00 sleep
0 R   201  7730  5369  0  78   0 -  1062 -      pts/0    00:00:00 ps
[rsimms@opus ~]$
```

Job Control
A feature of the bash shell

bg command

- Resumes a suspended job in the background

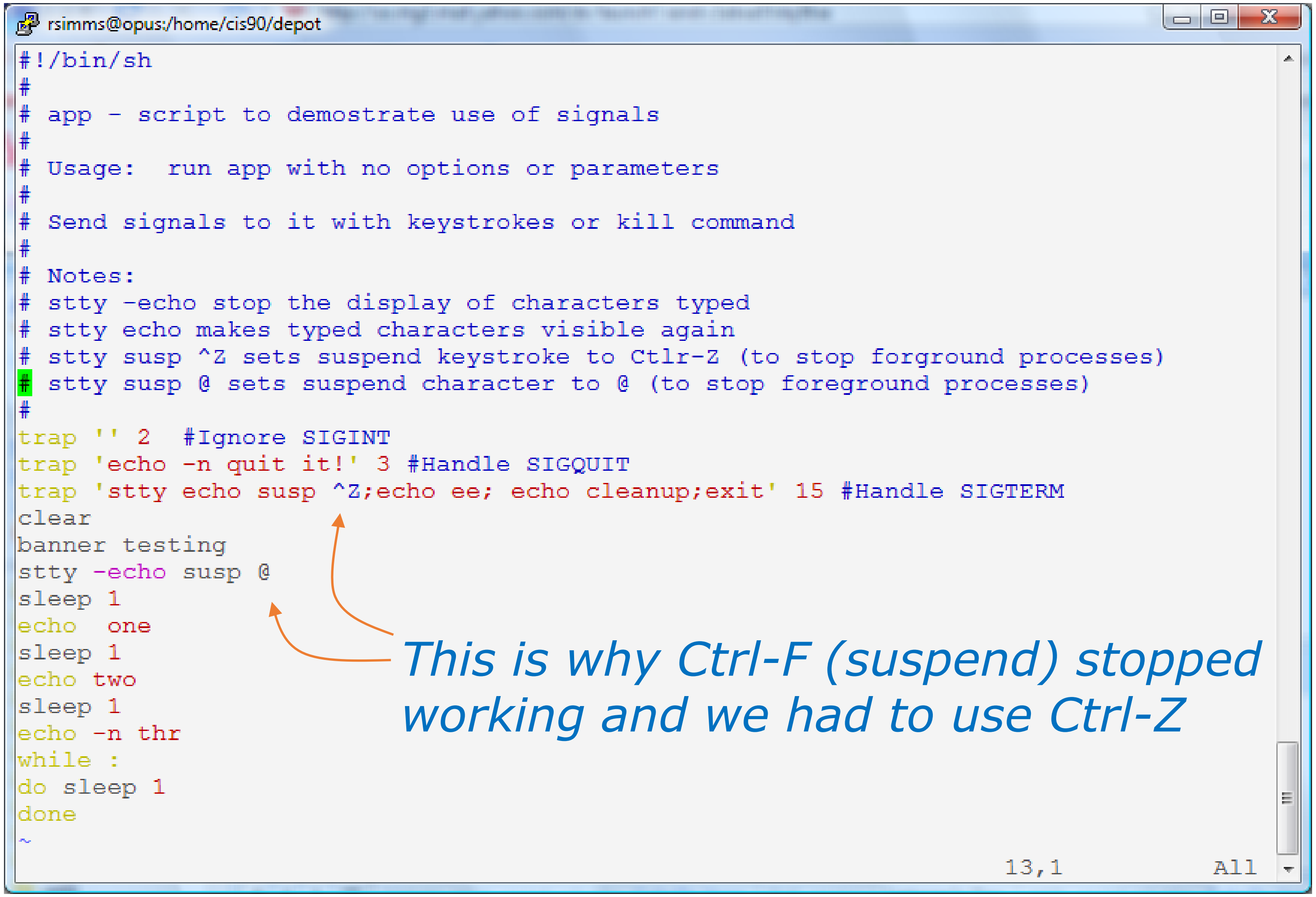
```
[rsimms@opus ~]$ sleep 5

[1]+  Stopped                 sleep 5
[rsimms@opus ~]$ bg
[1]+ sleep 5 &
[rsimms@opus ~]$
```

bg resumes the sleep command

PID 7728 is gone

```
[rsimms@opus ~]$ ps -l  -u rsimms
F S   UID   PID  PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
5 S   201  5368  5365  0  75   0 -  2460 -      ?        00:00:00 sshd
0 S   201  5369  5368  0  76   0 -  1165 wait   pts/0    00:00:00 bash
5 S   201  6203  6200  0  75   0 -  2491 -      ?        00:00:00 sshd
0 S   201  6204  6203  0  75   0 -  1165 -      pts/6    00:00:00 bash
0 R   201  7742  5369  0  78   0 -  1061 -      pts/0    00:00:00 ps
[rsimms@opus ~]$
```

Signals
Jim's app script


rsimms@opus:/home/cis90/depot

```
#!/bin/sh
#
# app - script to demostrate use of signals
#
# Usage:  run app with no options or parameters
#
# Send signals to it with keystrokes or kill command
#
# Notes:
# stty -echo stop the display of characters typed
# stty echo makes typed characters visible again
# stty susp ^Z sets suspend keystroke to Ctlr-Z (to stop forground processes)
# stty susp @ sets suspend character to @ (to stop foreground processes)
#
trap '' 2  #Ignore SIGINT
trap 'echo -n quit it!' 3 #Handle SIGQUIT
trap 'stty echo susp ^Z;echo ee; echo cleanup;exit' 15 #Handle SIGTERM
clear
banner testing
stty -echo susp @
sleep 1
echo  one
sleep 1
echo two
sleep 1
echo -n thr
while :
do sleep 1
done
~
                                                             13,1          All
```

This is why Ctrl-F (suspend) stopped working and we had to use Ctrl-Z

Tangent on bg and SIGCONT

Signals

What is signal 18?

Running process gets a signal

Signals

SIGSTKFLT	16	Stack fault
SIGCHLD	17	Child process has stopped or exited, changed (POSIX)
SIGCONT	18	Continue executing, if stopped (POSIX)
SIGSTOP	19	Stop executing(can't be caught or ignored) (POSIX)
SIGTSTP	20	Terminal stop signal (POSIX) ***Ctrl-Z or Ctrl-F***
SIGTTIN	21	Background process trying to read, from TTY (POSIX)
SIGTTOU	22	Background process trying to write, to TTY (POSIX)
SIGURG	23	Urgent condition on socket (4.2 BSD)
SIGXCPU	24	CPU limit exceeded (4.2 BSD)
SIGXFSZ	25	File size limit exceeded (4.2 BSD)
SIGVTALRM	26	Virtual alarm clock (4.2 BSD)
SIGPROF	27	Profiling alarm clock (4.2 BSD)
SIGWINCH	28	Window size change (4.3 BSD, Sun)
SIGIO	29	I/O now possible (4.2 BSD)
SIGPWR	30	Power failure restart (System V)

Signal 18 continues a stopped process … isn’t that what bg does?

The bg command is used to resume a stopped process

```
/home/cis90/roddyduk $ sleep 60
Ctrl-F (or Ctrl-Z) typed here
[1]+  Stopped                      sleep 60
/home/cis90/roddyduk $ bg
[1]+ sleep 60 &
/home/cis90/roddyduk $ jobs
[1]+  Running                      sleep 60 &
/home/cis90/roddyduk $ jobs
[1]+  Running                      sleep 60 &
/home/cis90/roddyduk $ jobs
[1]+  Done                         sleep 60
/home/cis90/roddyduk $
```

bg resumed the stopped process which runs till it is finished

*Instead of using **bg** to resume a stopped process in the background, lets try a SIGCONT (signal 18) instead*

```
/home/cis90/roddyduk $ sleep 60
Ctrl-F (or Ctrl-Z) typed here
[1]+  Stopped                 sleep 60
/home/cis90/roddyduk $ ps -l
F S   UID   PID  PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
0 S  1000 10705 10704  0  76   0 -  1165 wait   pts/0    00:00:00 bash
0 T  1000 10743 10705  0  75   0 -   926 finish pts/0    00:00:00 sleep
0 R  1000 10744 10705  0  78   0 -  1051 -      pts/0    00:00:00 ps
/home/cis90/roddyduk $ jobs
[1]+  Stopped                 sleep 60
/home/cis90/roddyduk $ kill -18 10743
/home/cis90/roddyduk $ jobs
[1]+  Running                 sleep 60 &
/home/cis90/roddyduk $ ps -l
F S   UID   PID  PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
0 S  1000 10705 10704  0  75   0 -  1165 wait   pts/0    00:00:00 bash
0 S  1000 10743 10705  0  85   0 -   926 322800 pts/0    00:00:00 sleep
0 R  1000 10746 10705  0  77   0 -  1050 -      pts/0    00:00:00 ps
/home/cis90/roddyduk $ jobs
[1]+  Running                 sleep 60 &
/home/cis90/roddyduk $ jobs
[1]+  Running                 sleep 60 &
/home/cis90/roddyduk $ jobs
[1]+  Done                    sleep 60
```

Note sending a 18 signal or using the bg command will resume a stopped process