Lesson Module Checklist

- Slides
- WB

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

- Lab tested and uploaded
- Tech file email for Lab 9 ready

- Materials uploaded
- Backup slides, CCC info, handouts on flash drive
- Check that backup room headset is charged
- Spare 9v battery for mic

Introductions and Credits

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

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

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

CIS 90 - Lesson 11

Instructor: **Rich Simms**
Dial-in: **888-450-4821**
Passcode: **761867**

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

Quiz

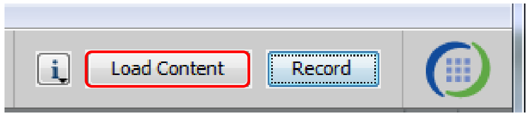
Please answer these questions **in the order** shown:

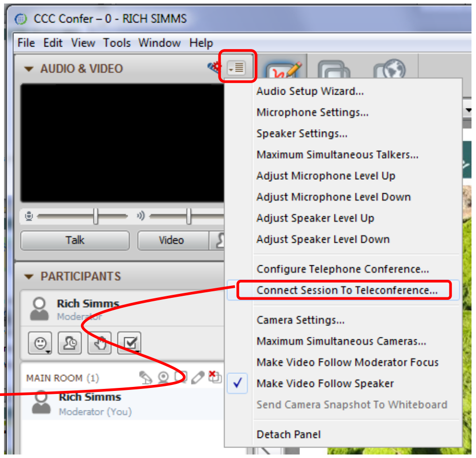
See electronic white board

email answers to: `risimms@cabrillo.edu`

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

[ ] **Preload White Board with *cis*lesson??*-WB***

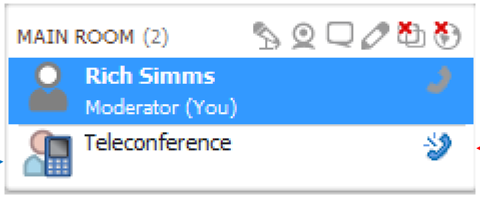




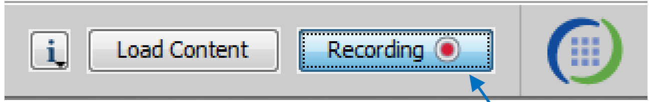
Audio Setup Wizard...
Microphone Settings...
Speaker Settings...
Maximum Simultaneous Talkers...
Adjust Microphone Level Up
Adjust Microphone Level Down
Adjust Speaker Level Up
Adjust Speaker Level Down
Configure Telephone Conference...
Connect Session To Teleconference...
Camera Settings...
Maximum Simultaneous Cameras...
Make Video Follow Moderator Focus
✓ Make Video Follow Speaker
Send Camera Snapshot To Whiteboard
Detach Panel

[ ] **Connect session to Teleconference**

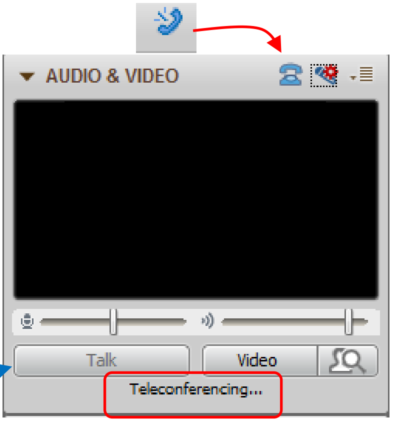
Session now connected to teleconference



[ ] **Is recording on?**



Red dot means recording



[ ] **Use teleconferencing, not mic**

Should be greyed out

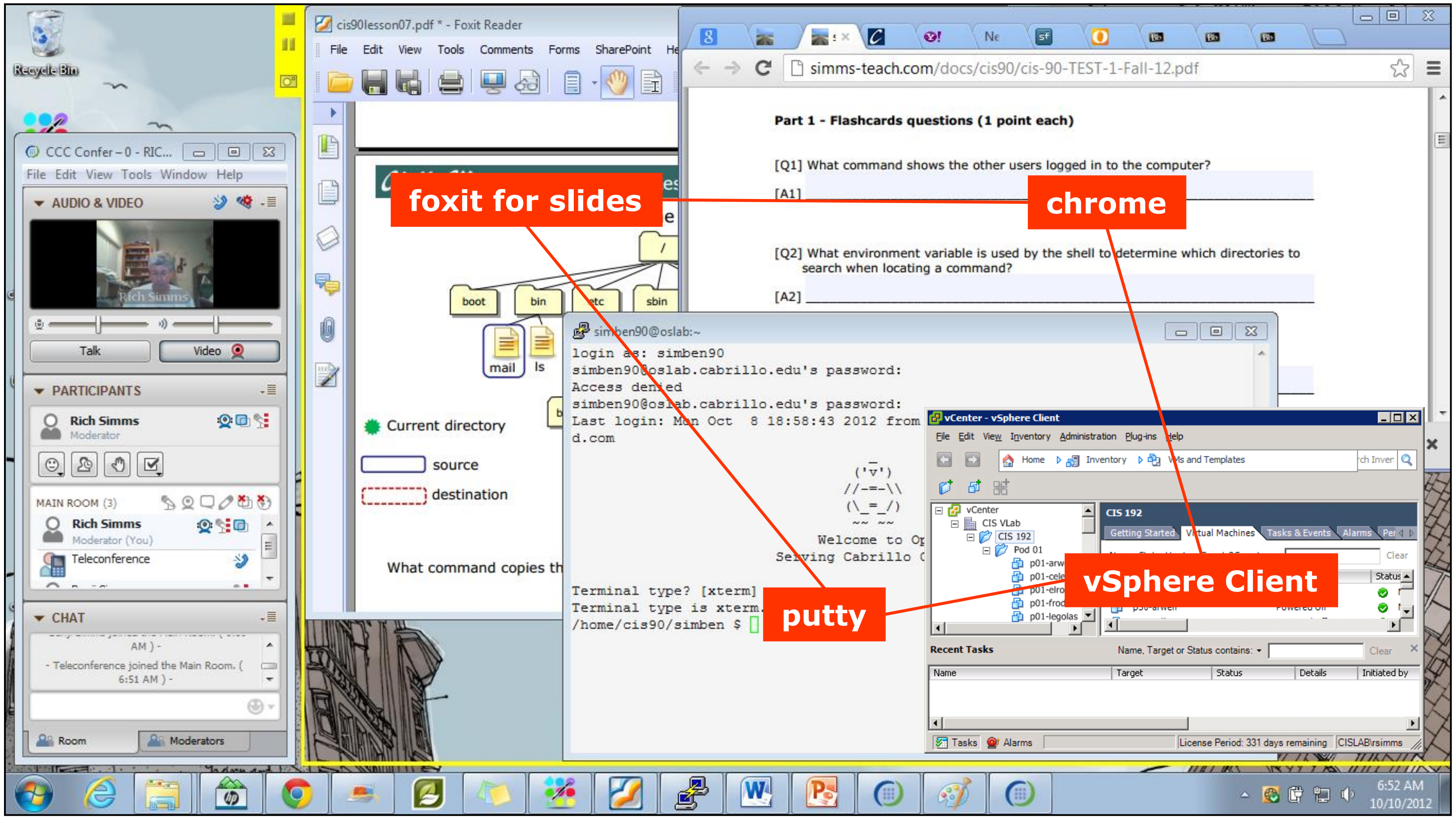
Cabrillo College est. 1959

CIS 90 - Lesson 11

CCC Confer

[ ] **Video (webcam) optional**

[ ] **layout and share apps**



Current directory

source

destination

What command copies th

Part 1 - Flashcards questions (1 point each)

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

[A1]

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

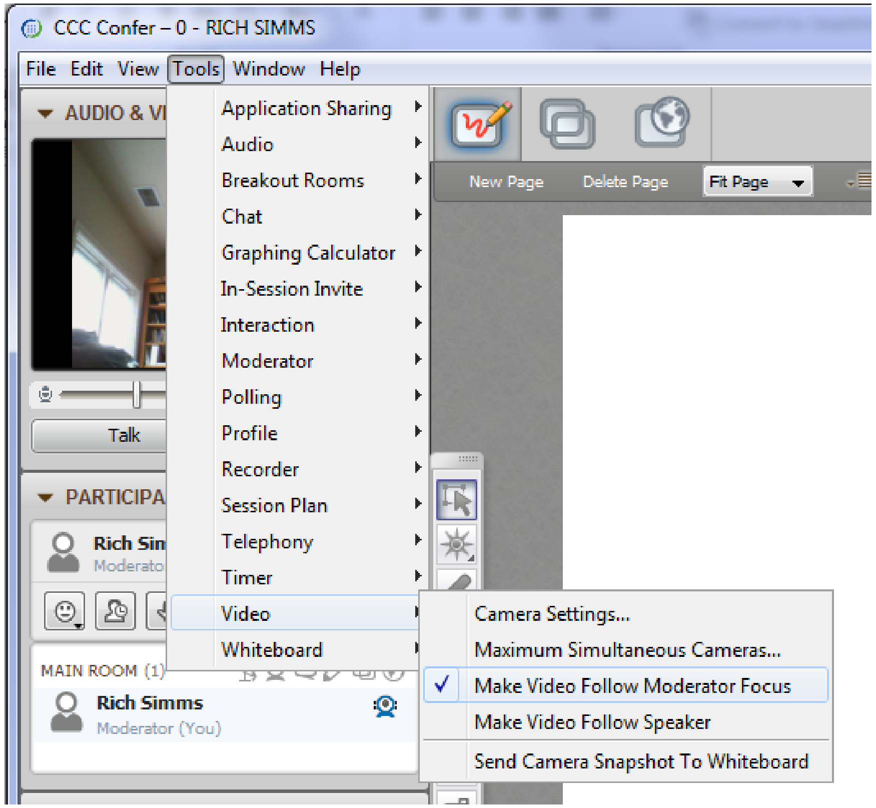
[A2]

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

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

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

[ ] Video (webcam) optional

[ ] Follow moderator

[ ] Double-click on postages stamps


CCC Confer – 0 - RICH SIMMS

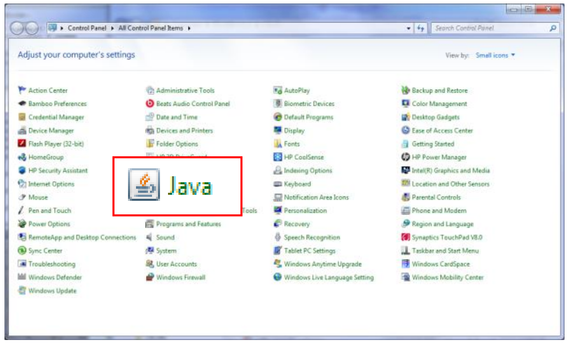
File Edit View Tools Window Help

✓ Make Video Follow Moderator Focus

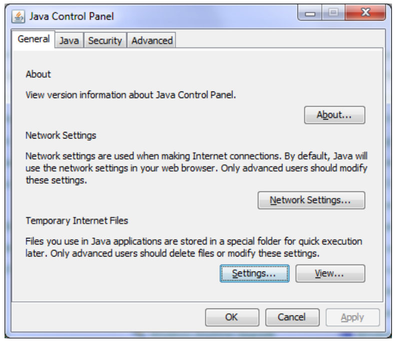
Universal Fix for CCC Confer:

1) Shrink (500 MB) and delete Java cache
2) Uninstall and reinstall latest Java runtime

Control Panel (small icons)



General Tab > Settings...



About

View version information about Java Control Panel.

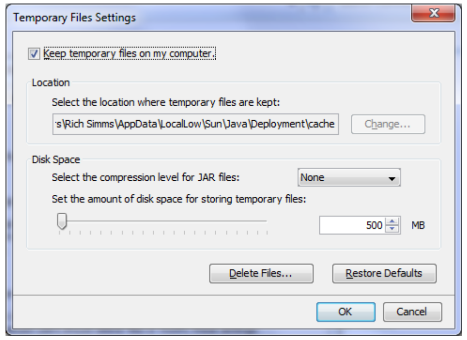
Network Settings

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

Temporary Internet Files

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

500MB cache size



[✓] Keep temporary files on my computer.

Location

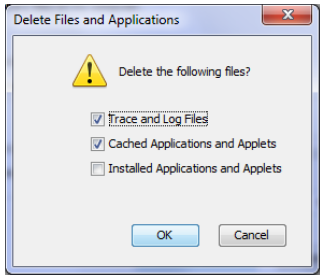
Select the location where temporary files are kept: s\Rich Simms\AppData\LocalLow\Sun\Java\Deployment\cache

Disk Space

Select the compression level for JAR files: None

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

Delete these



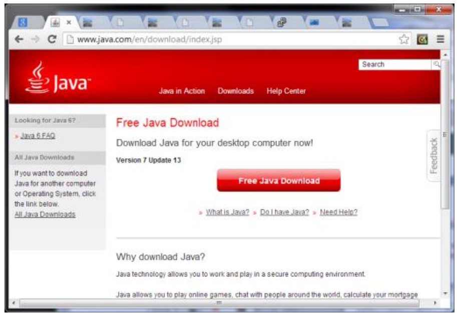
Delete the following files?

[✓] Trace and Log Files

[✓] Cached Applications and Applets

[ ] Installed Applications and Applets

Google Java download



Looking for Java 6?

» Java 6 FAQ

All Java Downloads

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

All Java Downloads

Free Java Download

Download Java for your desktop computer now!

Version 7 Update 13

Why download Java?

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

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

vi editor

Objectives	Agenda
• Create and modify text files	• Quiz • Questions from last week • more on grep • Review on processes • The vi editor • Wrap up

Questions

Questions

Lesson material?

Labs?

How this course works?

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

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.*

Housekeeping

Previous material and assignment

1. Questions?

2. Lab 8 due tonight

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

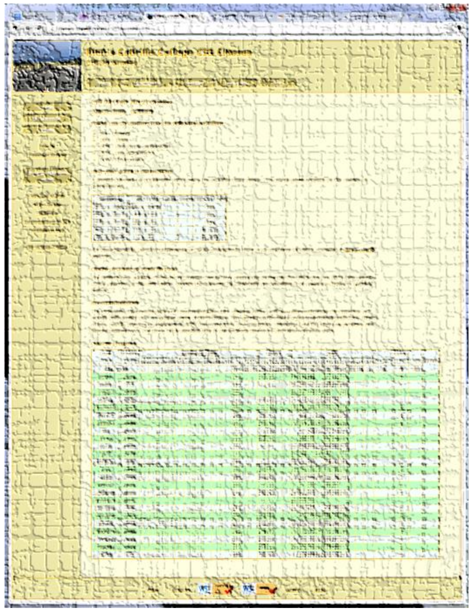
Don't wait till midnight tonight to see if this worked! Test with an earlier time.

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

4. You can still send me your photo for our class page if you want 3 points extra credit

Managing your grade

Use the web page



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

Using Jesse's checkgrades script

```
adaldrida: 71% (238 of 331 points)
anborn: 101% (337 of 331 points)
arador: 46% (154 of 331 points)
aragorn: 64% (213 of 331 points)
balrog: 0% (0 of 331 points)
bilbo: 90% (300 of 331 points)
bombadil: 8% (28 of 331 points)
celebrian: 62% (206 of 331 points)
cirdan: 54% (179 of 331 points)
durin: 87% (288 of 331 points)
dwalin: 91% (302 of 331 points)
elrond: 105% (350 of 331 points)
eomer: 101% (337 of 331 points)
faramir: 108% (359 of 331 points)
frodo: 100% (333 of 331 points)
gimli: 61% (204 of 331 points)
goldberry: 85% (284 of 331 points)
gwaihir: 69% (231 of 331 points)
haldir: 61% (205 of 331 points)
ingold: 90% (299 of 331 points)
ioreth: 95% (315 of 331 points)
legolas: 103% (341 of 331 points)
marhari: 91% (304 of 331 points)
pallando: 85% (282 of 331 points)
quickbeam: 51% (171 of 331 points)
samwise: 96% (319 of 331 points)
sauron: 93% (310 of 331 points)
shadowfax: 83% (278 of 331 points)
strider: 103% (344 of 331 points)
theoden: 99% (329 of 331 points)
treebeard: 95% (316 of 331 points)
tulkas: 91% (302 of 331 points)
```

As of May 1, 2013

Managing your grade

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

Points gone by
- 7 quizzes - 21 points
- 2 tests - 60 points
- 2 forum periods - 40 points
- 7 labs - 210 points

331 points

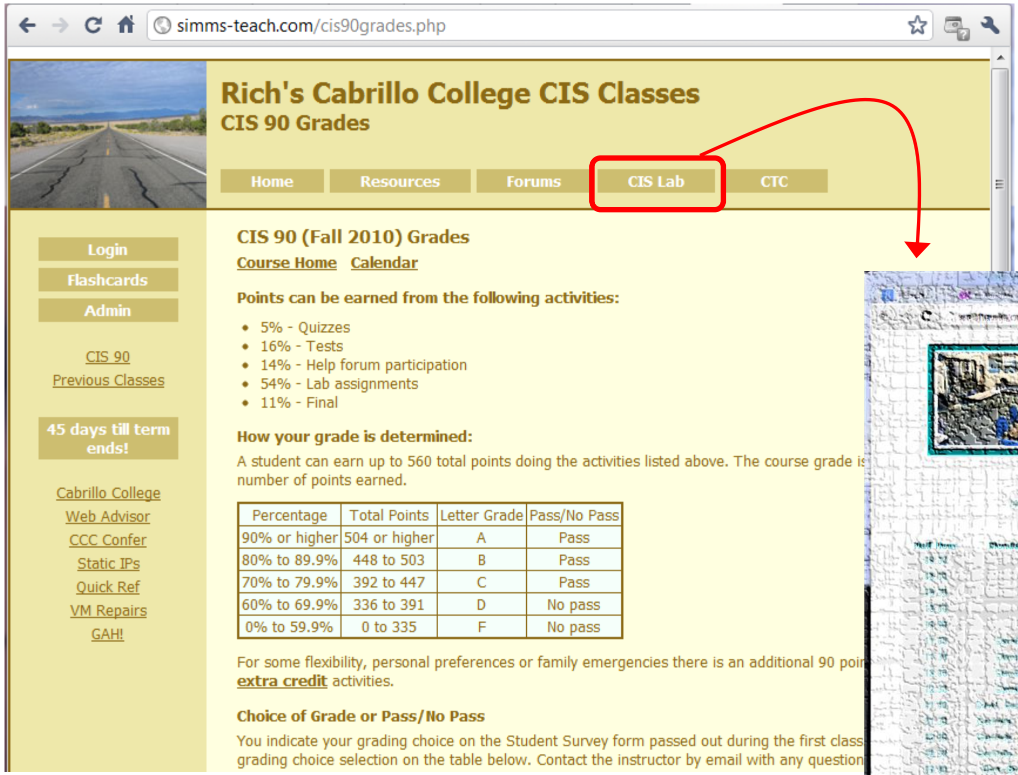
Points yet to earn
- 3 quizzes - 9 points
- 1 test - 30 points
- 2 forum periods - 40 points
- 3 labs - 90 points
- 1 final project - 60 points

229 points

- Plus extra credit - up to 90 points

Managing your grade

Getting extra help for CIS 90



simms-teach.com/cis90grades.php

Rich's Cabrillo College CIS Classes
CIS 90 Grades

CIS 90
Previous Classes

45 days till term ends!

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

CIS 90 (Fall 2010) Grades

Course Home Calendar

Points can be earned from the following activities:

- 5% - Quizzes
- 16% - Tests
- 14% - Help forum participation
- 54% - Lab assignments
- 11% - Final

How your grade is determined:

A student can earn up to 560 total points doing the activities listed above. The course grade is number of points earned.

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

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

Choice of Grade or Pass/No Pass

You indicate your grading choice on the Student Survey form passed out during the first class grading choice selection on the table below. Contact the instructor by email with any question

Come by the lab and get help from instructors and student assistants

Cabrillo Network & Systems Technology Lab

Fall 2012 Instructor and Lab Assistant Hours

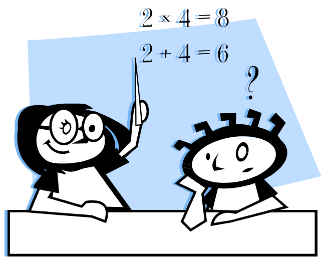
[illegible]

Managing your grade

Getting extra help for CIS 90

- Rich’s Office Hours in Room 2501 (right after class) or TBA (contact me)

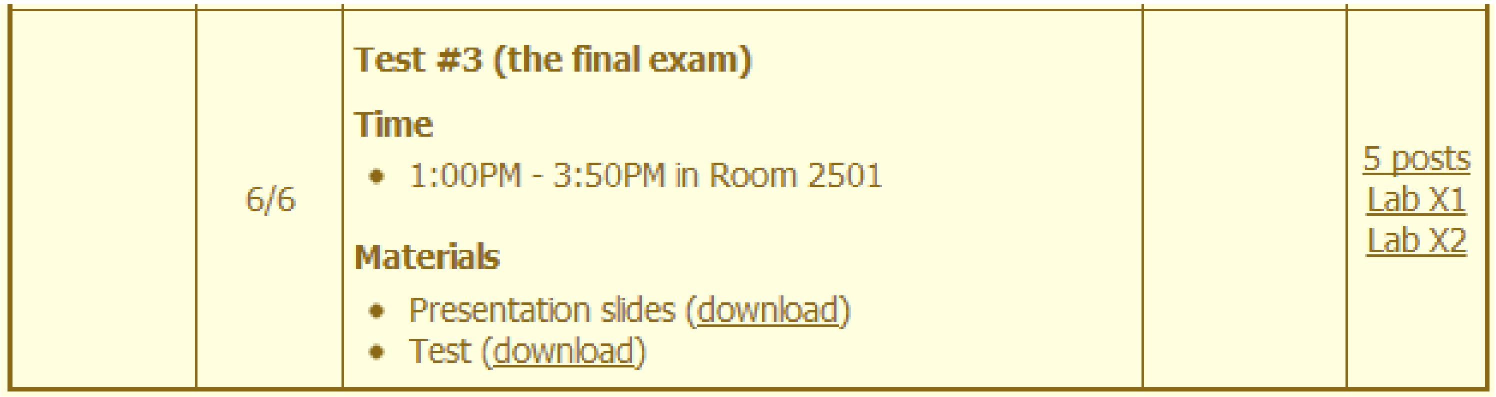
- Ask questions on the Forum at:
  http://oslab.cabrillo.edu/forum/



Final Exam

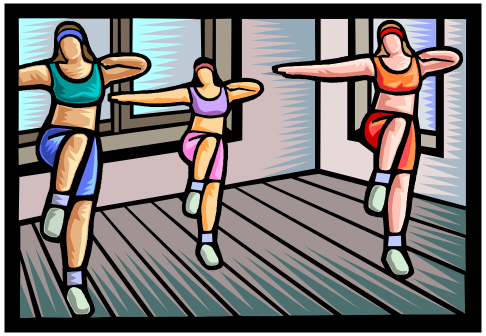
Test #3 (final exam)

- Must be face-to-face or proctored (not online using CCC Confer).
- We will be in room 2501 on campus.



	6/6	**Test #3 (the final exam)** **Time** • 1:00PM - 3:50PM in Room 2501 **Materials** • Presentation slides (download) • Test (download)		5 posts Lab X1 Lab X2

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 portion (or all) 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      6251     1  0 Jul31 ?        00:00:04 cupsd -C /etc/cups/cupsd.conf
simben90 27027 26966  0 08:47 pts/3    00:00:00 grep cups
```

Yes it is, with PID=6251

grep practice

- Is the cronjob daemon (crond) running right now?
- 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-2.2.15-15.el6.centos.1.i686
httpd-2.2.15-15.el6.centos.1.i686
httpd-manual-2.2.15-15.el6.centos.1.noarch
```

Yes, version 2.2.15 has been installed

grep practice

- Has the mysql-server package been installed on Opus?
- If installed on Opus, type the version of mysql in the chat window

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/0        50-0-68-235.dsl. Mon Apr 23 05:39   still logged in
simben90 pts/6        10.64.25.2       Wed Apr 18 12:48 - 16:51  (04:02)
simben90 pts/5        10.64.25.2       Wed Apr 18 12:48 - 16:51  (04:02)
simben90 pts/4        10.64.25.2       Wed Apr 18 12:48 - 16:51  (04:03)
simben90 pts/1        50-0-68-235.dsl. Wed Apr 18 09:06 - 10:23  (01:17)
```

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

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

Similar to lab 8. This is how to show which shell uses the most memory when it runs as a process and record that answer in a file

```
[rsimms@oslab ~]$ ls /bin/*sh
/bin/bash  /bin/csh  /bin/dash  /bin/ksh  /bin/rbash  /bin/sh  /bin/tcsh

[rsimms@oslab ~]$ ksh
$ dash
$ sh
sh-4.1$ csh

[rsimms@oslab ~]$ ps -l
F S   UID   PID  PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
0 S   201 27553 27552  0  80   0 -  1308 -      pts/0    00:00:00 bash
0 S   201 27651 27553  0  80   0 -  1376 -      pts/0    00:00:00 ksh
0 S   201 27652 27651  0  80   0 -   517 -      pts/0    00:00:00 dash
0 S   201 27653 27652  0  80   0 -  1307 -      pts/0    00:00:00 sh
0 S   201 27654 27653  0  80   0 -  1458 -      pts/0    00:00:00 csh
0 R   201 27663 27654  0  80   0 -  1214 -      pts/0    00:00:00 ps

[rsimms@oslab ~]$ ps -l | grep csh
0 S   201 27654 27653  0  80   0 -  1458 -      pts/0    00:00:00 csh

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

[rsimms@oslab ~]$ cat bigshell
0 S   201 27654 27653  0  80   0 -  1458 -      pts/0    00:00:00 csh
```

grep practice

- For the bash, dash, ksh, sh and csh shells, which shell process uses the least memory?
- What command that would redirect the line of output for the command using the least amount of memory to the file *smallshell*
- Type the command you used and its output into the chat window

grep usage – search inside files

How many CIS 90 user accounts are there?

```
/home/cis90/simben $ grep cis90 /etc/passwd | wc -l
29
```

There are 29

grep practice

- How many CIS 172 accounts are there on Opus?
- Type the number of CIS 172 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

Where does the PS1 "prompt" variable get set?

```
/home/cis90/simben $ grep -R "PS1=" /etc/bash* $HOME 2> /dev/null
/etc/bash_completion.d/git:#        PS1='[\u@\h \W$(__git_ps1 " (%s)")]\$ '
/etc/bashrc:  [ "$PS1" = "\\s-\\v\\\$ " ] && PS1="[\u@\h \W]\\$ "
/etc/bashrc:  #   PS1="[\u@\h:\l \W]\\$ "
/home/cis90/simben/class/labs/lab04.graded:21) PS1='$PWD $ '
/home/cis90/simben/class/exams/test01.graded:(A32) PS1='\d $ '
/home/cis90/simben/.bash_profile:PS1='$PWD $ '
/home/cis90/simben/lab04.graded:21) PS1='$PWD $ '
/home/cis90/simben/test01.graded:(A32) PS1='\d $ '
```

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.

grep practice

- Find the file in the /usr/lib portion of the file tree that contains "hot pototo dance" (yes, potato is misspelled).
- Type the absolute pathname of the file in the chat window.

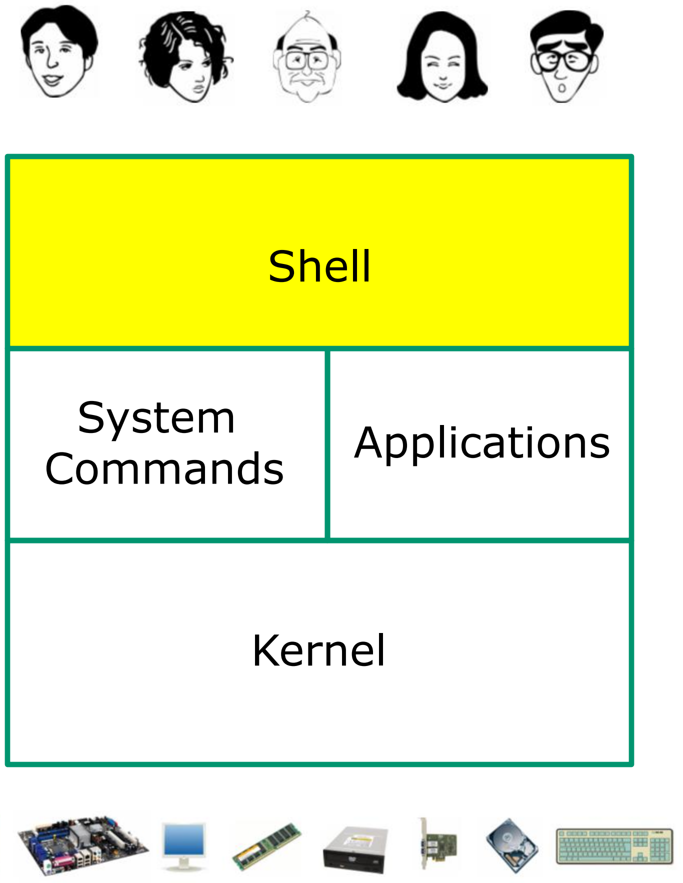
Shell six steps (REVIEW)

Example Command

```
/home/cis90/simben $ find / -name *egg 2> /dev/null
/home/cis90/lovben/1968.egg
/home/cis90/wismar/basket/1968.egg
/home/cis90/perste/basket/1968.egg
/home/cis90/perste/1968.egg
/home/cis90/cis/1968.egg
/home/cis90/paljay/basket/1968.egg
/home/cis90/paljay/1968.egg
/home/cis90/fareli/1968.egg
/home/cis90/rodduk/1968.egg
/home/cis90/wiltyr/basket/1968.egg
/home/cis90/wiltyr/1968.egg
 < snipped >
/home/cis90/mennat/1968.egg
/home/cis90/berric/basket/1968.egg
/home/cis90/berric/1968.egg
/home/cis90/goljor/1968.egg
/home/cis90/marand/1968.egg
/home/cis90/lejmic/basket/1968.egg
/home/cis90/davmic/basket/1968.egg
/home/cis90/davmic/1968.egg
/home/cis90/schrya/Basket/1968.egg
/home/cis90/simben $
```

On the next slides we will walk through each of the six steps the shell performs for this command

Prompt Step



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

Prompt Step

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

The shell prompt is output from the bash shell program directed to your terminal device

- Benji is using the bash shell. There are many other shells such as sh, ksh and csh. The last field in the line for his account in */etc/passwd* determines the shell that is run when he logs in.
- The bash program resides in the */bin* directory
- The command prompt appearance is defined by the PS1 variable. You can output a prompt yourself using **echo $PS1**

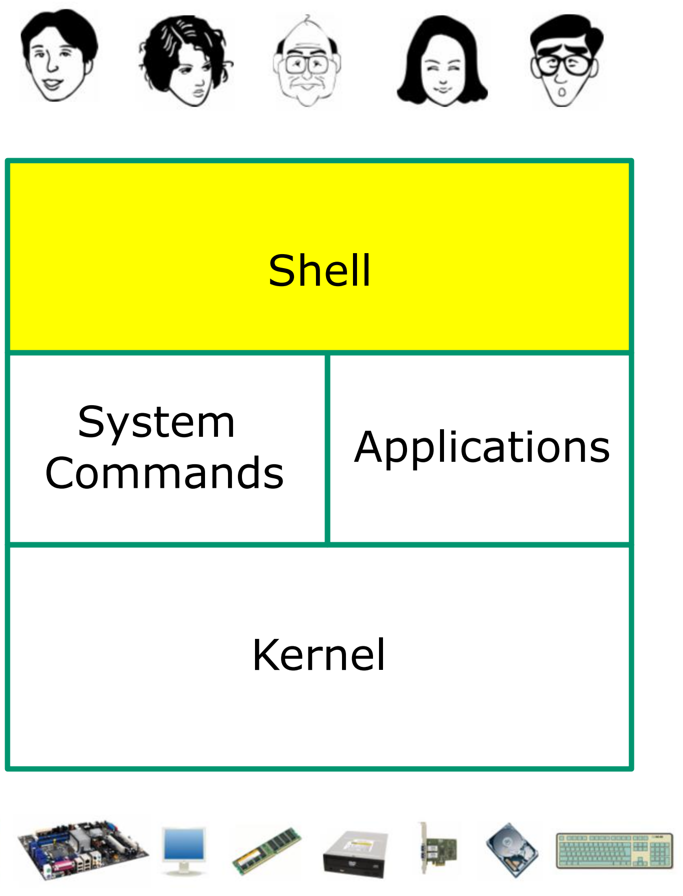
```
/home/cis90/simben $ grep $LOGNAME /etc/passwd
simben90:x:1001:190:Benji Simms:/home/cis90/simben:/bin/bash

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

Prompt Step

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

Benji types in this find command in response to the shell prompt

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 *egg 2> /dev/null
```

The shell must expand filename expansion characters and variables during the parse step. Note there is an invisible <newline> metacharacter at the end of the command

Parsing RESULTS:

Command: **find**

Options and arguments:
/
-name
1968.egg

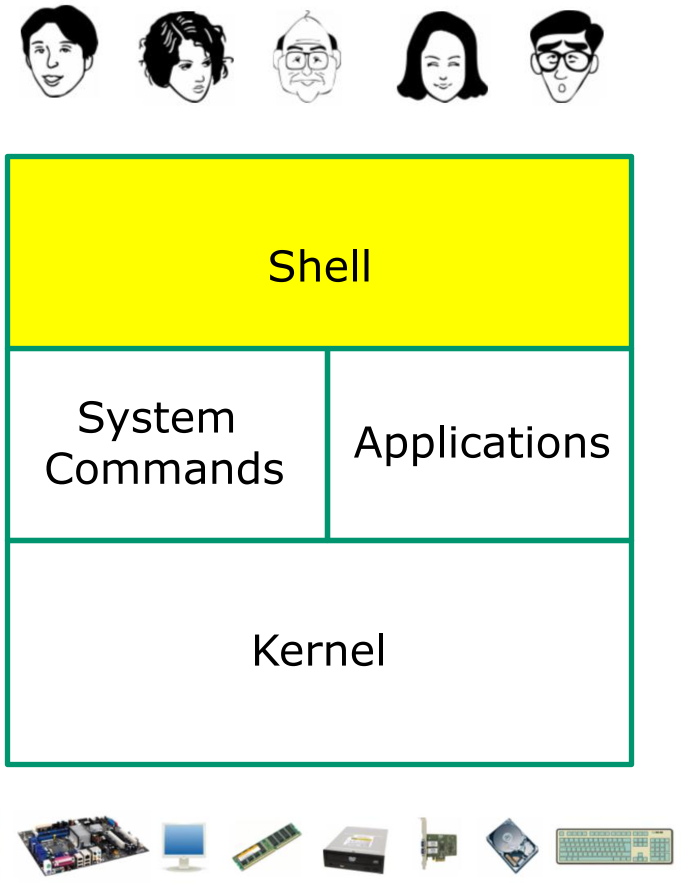
This will be passed to the command (if the command can be located on the path)

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

Command: **find**

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

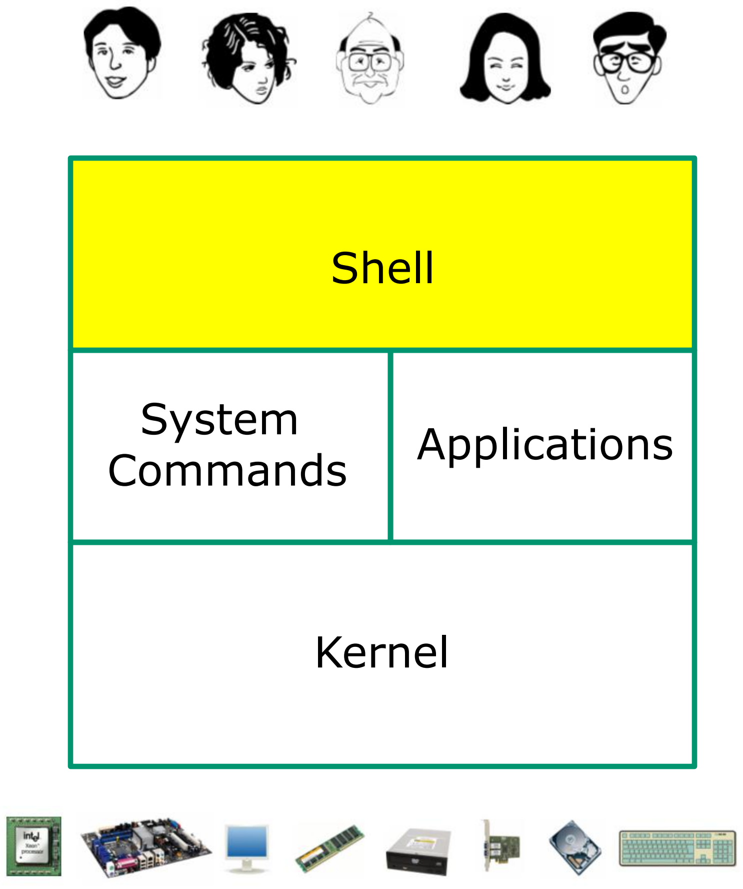
Benji's path is defined by the value of his PATH variable

1st directory searched: /usr/lib/qt-3.3/bin
2nd directory searched: /usr/local/bin
3rd directory searched: **/bin**
4th directory searched: /usr/bin
5th directory searched: /usr/local/sbin
6th directory searched: /usr/sbin
7th directory searched: /sbin
8th directory searched: /home/cis90/simben/../bin
9th directory searched: /home/cis90/simben/bin
10th directory searched: .

The shell locates the find command in the /bin directory

```
/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:.
```

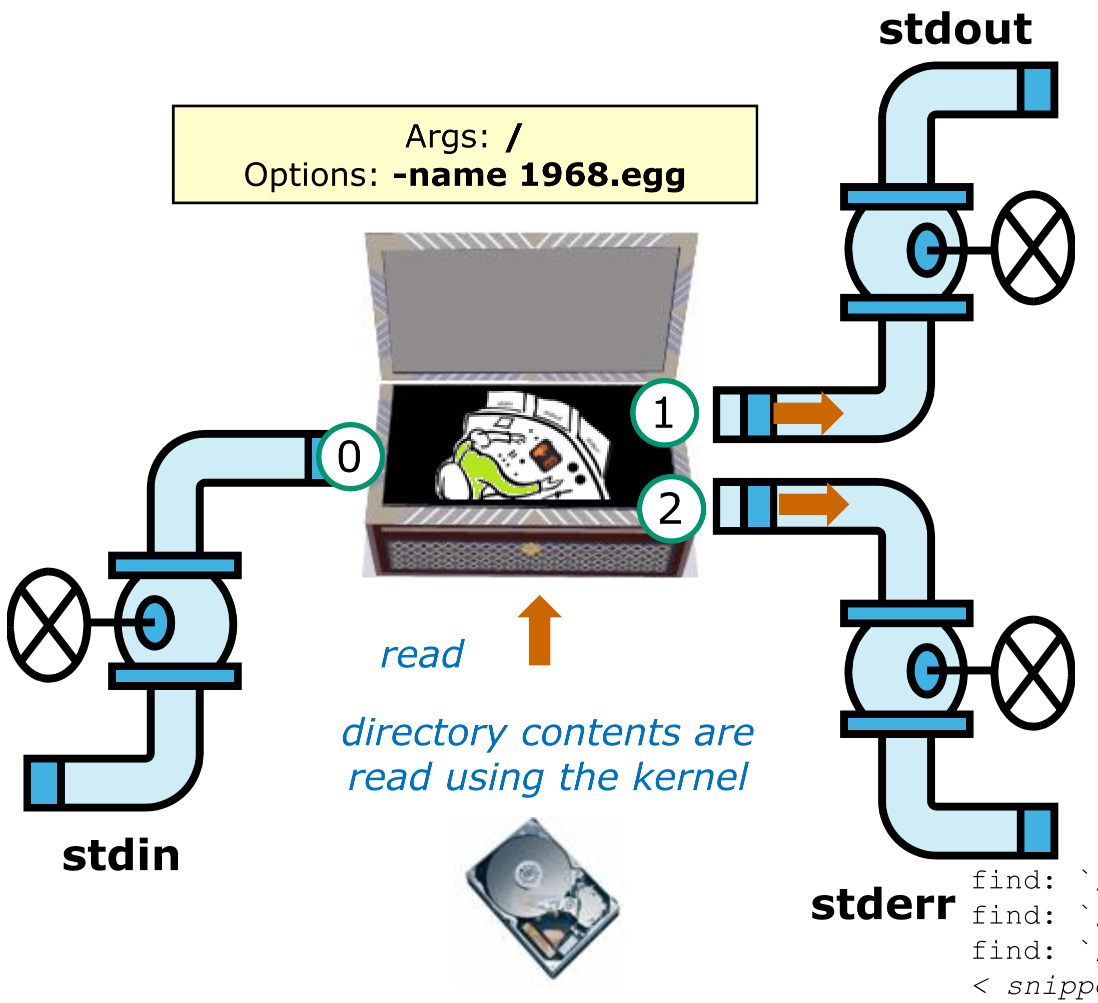
Execute Step



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

Execute Step

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

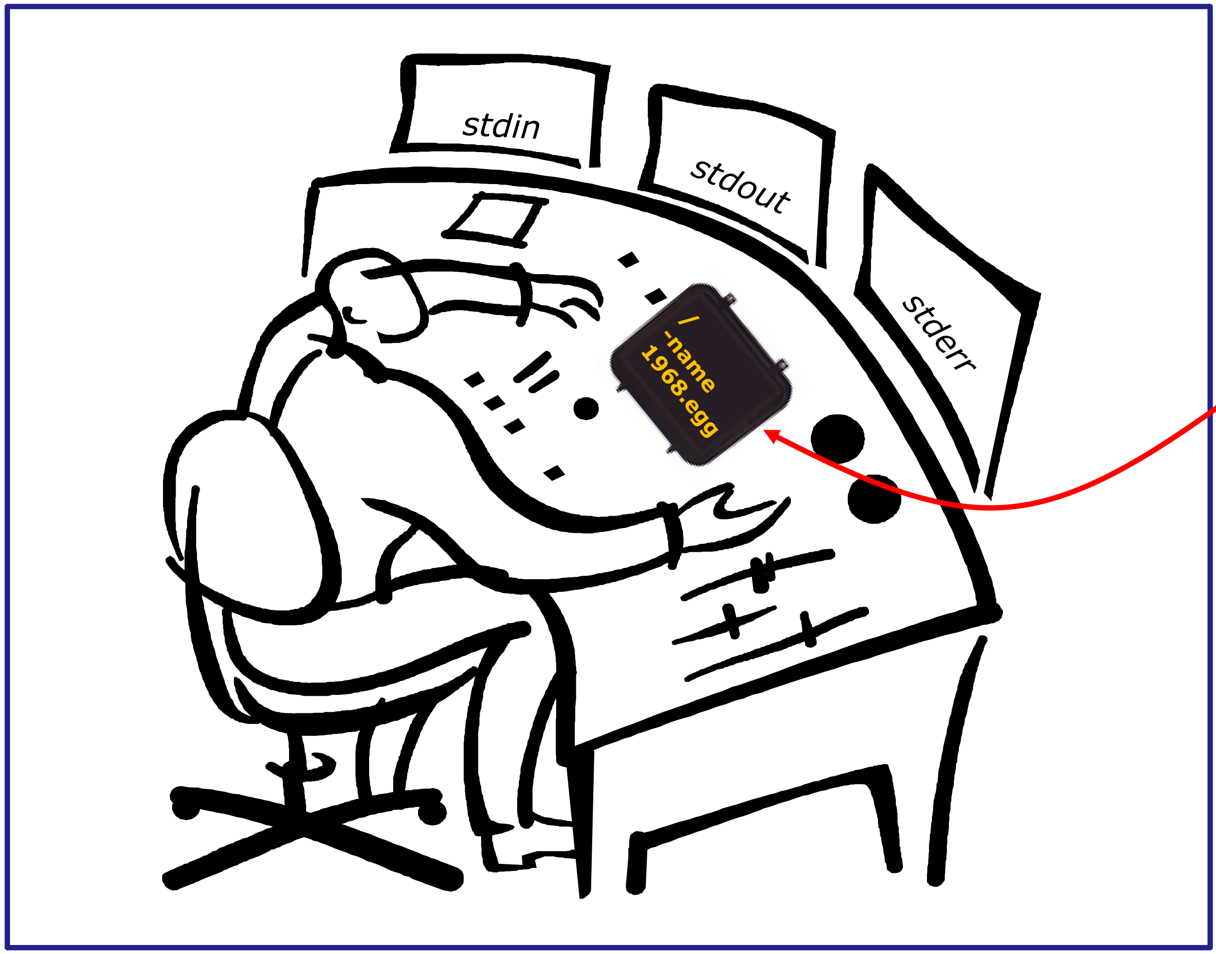


```
/home/cis90/lovben/1968.egg
/home/cis90/wismar/basket/1968.egg
/home/cis90/perste/basket/1968.egg
/home/cis90/perste/1968.egg
/home/cis90/cis/1968.egg
/home/cis90/paljay/basket/1968.egg
< snipped >
```

/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



A **process:**

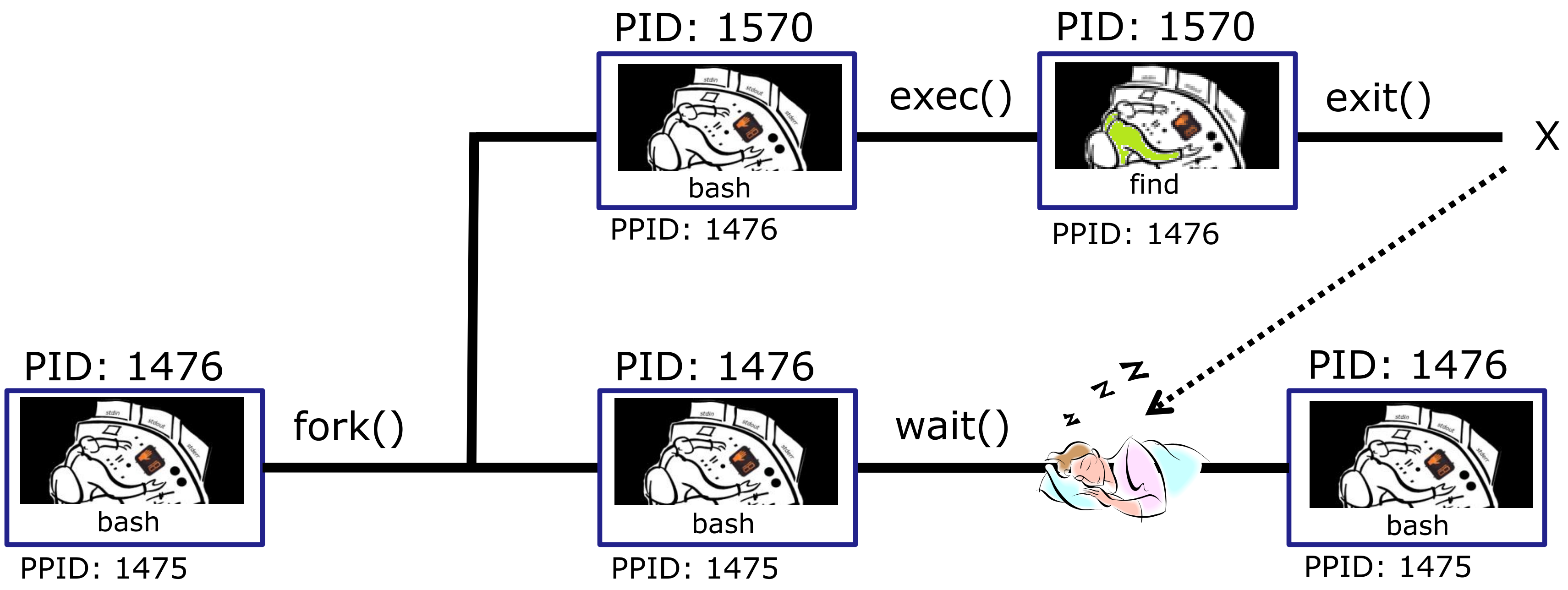
- Is provided with parsed & expanded options and arguments from the shell

- 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**

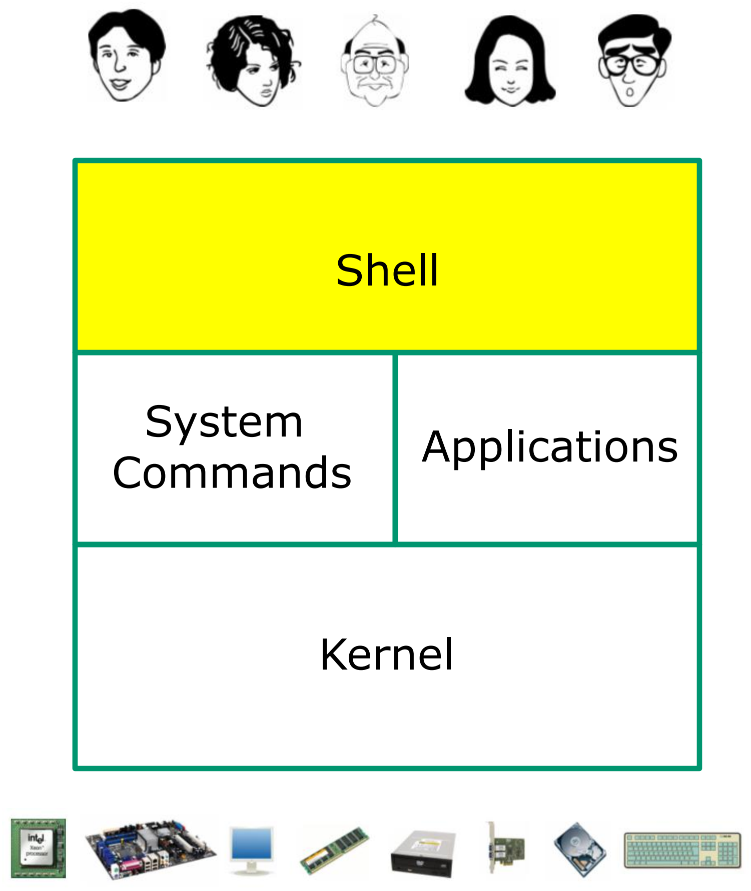
The find process only sees what the shell gives it. It does not see what the user typed!

Execute Step



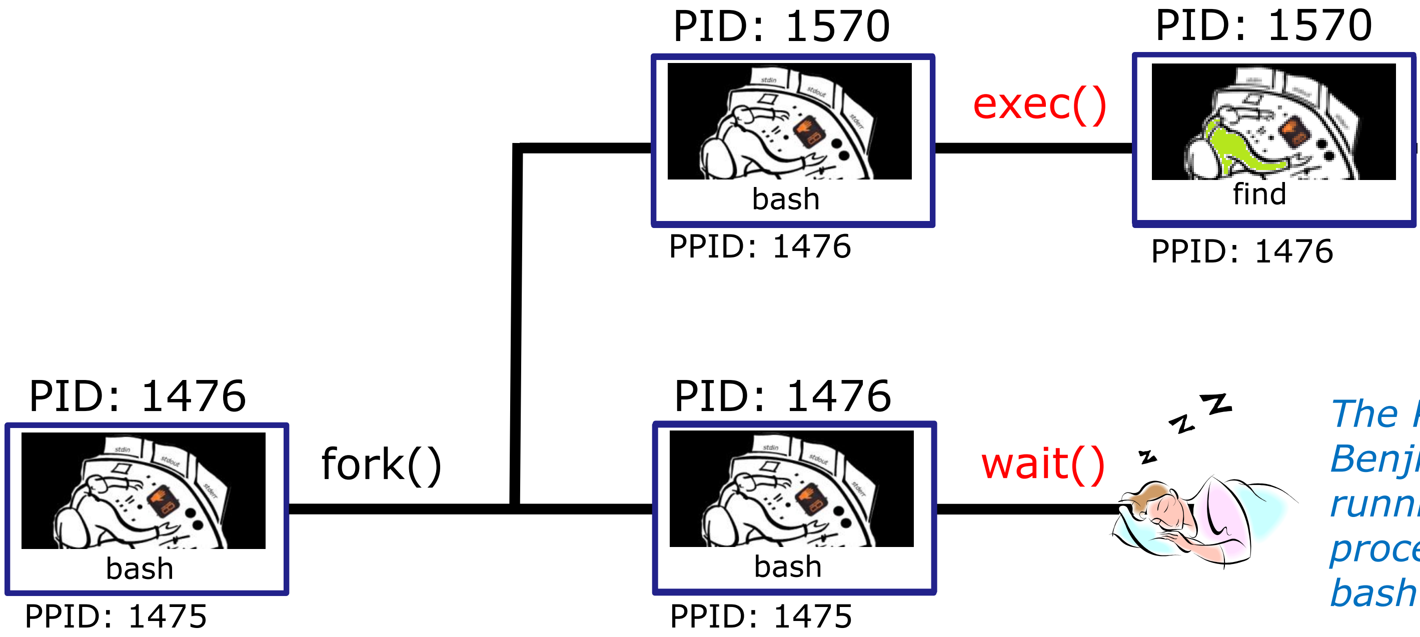
*bash executes the find command by cloning itself with a **fork()** system call to create a new child process. With an **exec()** system call, the new child process is overlaid with the find code instructions. bash sleeps by making a **wait()** system call while the find child process runs. The child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

Nap Step



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

Nap Step

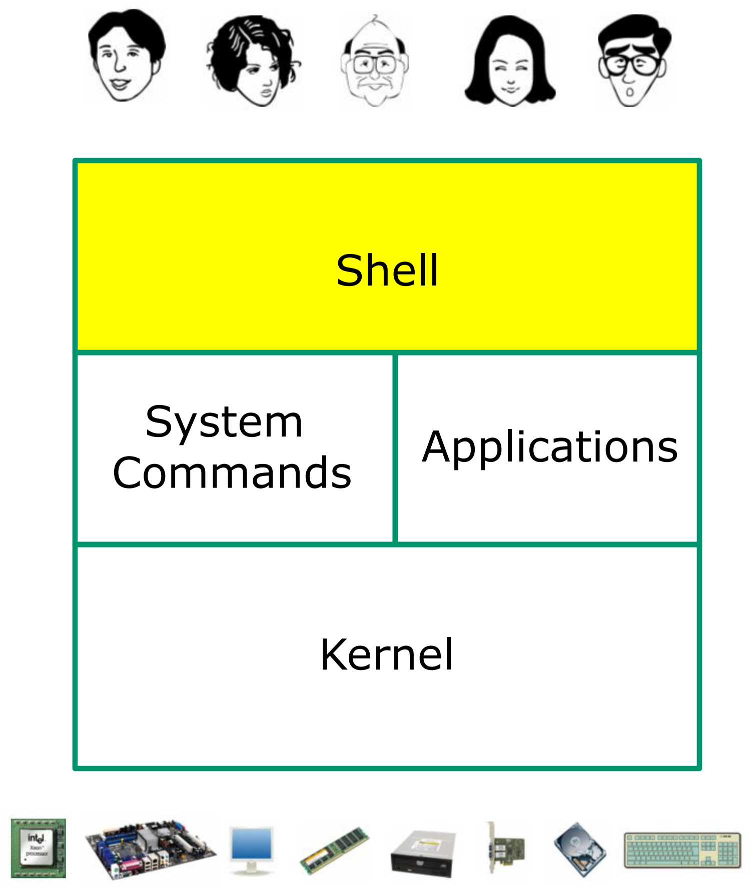


*The PS command shows Benji's **find** command is running as a child process while the parent bash shell sleeps*

```
[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
```

R=Running (PID 1570 find), S=Sleeping (PID 1476 bash)

Repeat Step

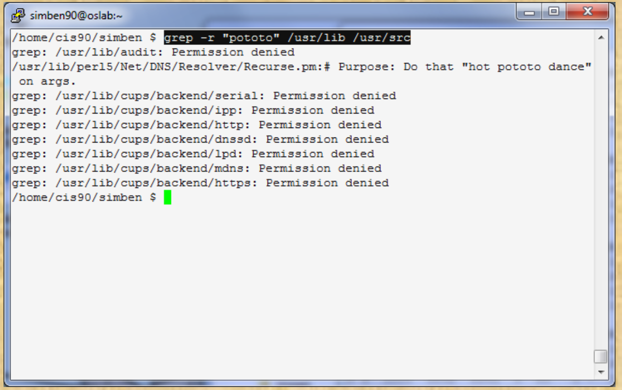


1) Prompt
2) Parse
3) Search
4) Execute
5) Nap
6) Repeat

Process activity

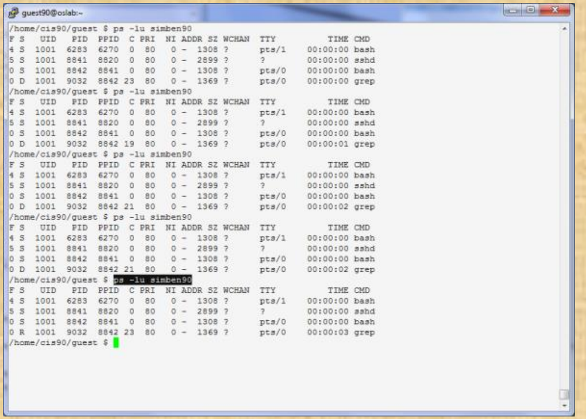
- See if you can do a **ps** command that illustrates what happens when a user runs a long **grep** command.
- The **ps** output should show "parent" bash S=Sleeping while the "child" **grep** command is either R=Running or in D=Uninterruptible sleep (IO)
- Start a second login session to observe your processes
- Write your grep PID and status into the chat window when done

```
/home/cis90/simben $ grep -r "pototo" /usr/lib /usr/src
```



```
/home/cis90/simben $ grep -r "pototo" /usr/lib /usr/src
grep: /usr/lib/audit: Permission denied
/usr/lib/perl5/Net/DNS/Resolver/Recurse.pm:# Purpose: Do that "hot pototo dance"
 on args.
grep: /usr/lib/cups/backend/serial: Permission denied
grep: /usr/lib/cups/backend/ipp: Permission denied
grep: /usr/lib/cups/backend/http: Permission denied
grep: /usr/lib/cups/backend/dnssd: Permission denied
grep: /usr/lib/cups/backend/lpd: Permission denied
grep: /usr/lib/cups/backend/mdns: Permission denied
grep: /usr/lib/cups/backend/https: Permission denied
/home/cis90/simben $
```

```
/home/cis90/guest $ ps -lu simben90
```



```
/home/cis90/guest $ ps -lu simben90
F S   UID   PID  PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
4 S  1001  6283  6270  0  80   0 -  1308 ?      pts/1    00:00:00 bash
5 S  1001  8841  8820  0  80   0 -  2899 ?      ?        00:00:00 sshd
0 S  1001  8842  8841  0  80   0 -  1308 ?      pts/0    00:00:00 bash
0 D  1001  9032  8842 23  80   0 -  1369 ?      pts/0    00:00:00 grep
/home/cis90/guest $ ps -lu simben90
F S   UID   PID  PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
4 S  1001  6283  6270  0  80   0 -  1308 ?      pts/1    00:00:00 bash
5 S  1001  8841  8820  0  80   0 -  2899 ?      ?        00:00:00 sshd
0 S  1001  8842  8841  0  80   0 -  1308 ?      pts/0    00:00:00 bash
0 D  1001  9032  8842 19  80   0 -  1369 ?      pts/0    00:00:01 grep
/home/cis90/guest $ ps -lu simben90
F S   UID   PID  PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
4 S  1001  6283  6270  0  80   0 -  1308 ?      pts/1    00:00:00 bash
5 S  1001  8841  8820  0  80   0 -  2899 ?      ?        00:00:00 sshd
0 S  1001  8842  8841  0  80   0 -  1308 ?      pts/0    00:00:00 bash
0 D  1001  9032  8842 21  80   0 -  1369 ?      pts/0    00:00:02 grep
/home/cis90/guest $ ps -lu simben90
F S   UID   PID  PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
4 S  1001  6283  6270  0  80   0 -  1308 ?      pts/1    00:00:00 bash
5 S  1001  8841  8820  0  80   0 -  2899 ?      ?        00:00:00 sshd
0 S  1001  8842  8841  0  80   0 -  1308 ?      pts/0    00:00:00 bash
0 D  1001  9032  8842 21  80   0 -  1369 ?      pts/0    00:00:02 grep
/home/cis90/guest $ ps -lu simben90
F S   UID   PID  PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
4 S  1001  6283  6270  0  80   0 -  1308 ?      pts/1    00:00:00 bash
5 S  1001  8841  8820  0  80   0 -  2899 ?      ?        00:00:00 sshd
0 S  1001  8842  8841  0  80   0 -  1308 ?      pts/0    00:00:00 bash
0 R  1001  9032  8842 23  80   0 -  1369 ?      pts/0    00:00:03 grep
/home/cis90/guest $
```

Review of Signals

Signals

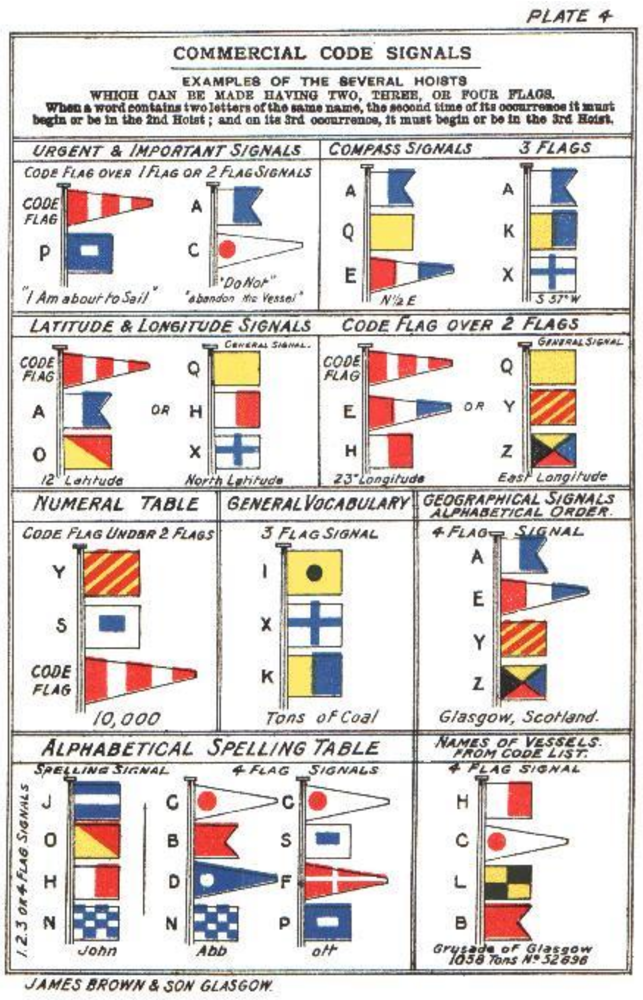
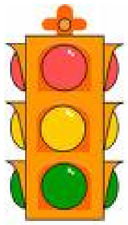
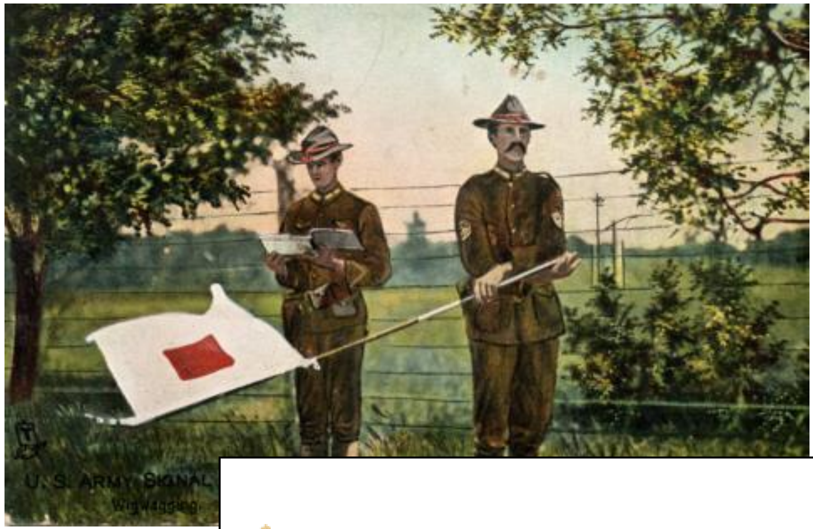
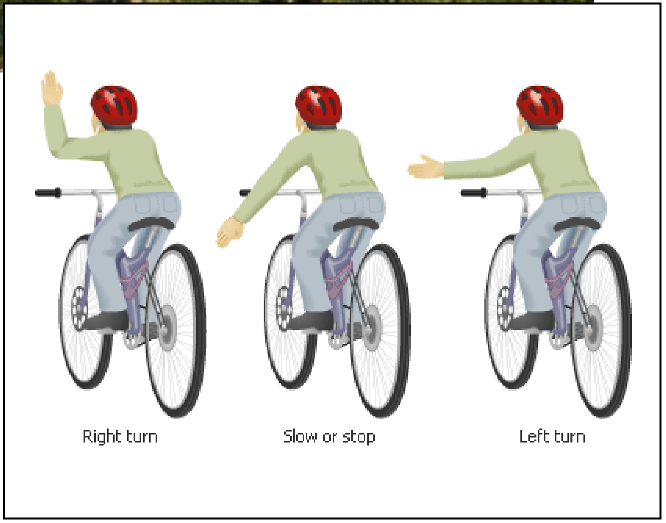


PLATE 4
COMMERCIAL CODE SIGNALS
EXAMPLES OF THE SEVERAL HOISTS
WHICH CAN BE MADE HAVING TWO, THREE, OR FOUR FLAGS.
When a word contains two letters of the same name, the second time of its occurrence it must begin or be in the 2nd Hoist; and on its 3rd occurrence, it must begin or be in the 3rd Hoist.
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
GENERAL SIGNAL
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

20


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**

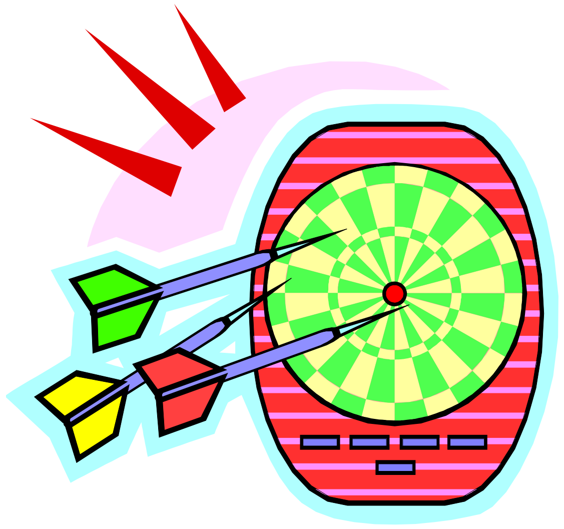
- Where # is the signal number and PID is the process id.
- if no number is specified, SIGTERM (-15) is sent.

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 …*

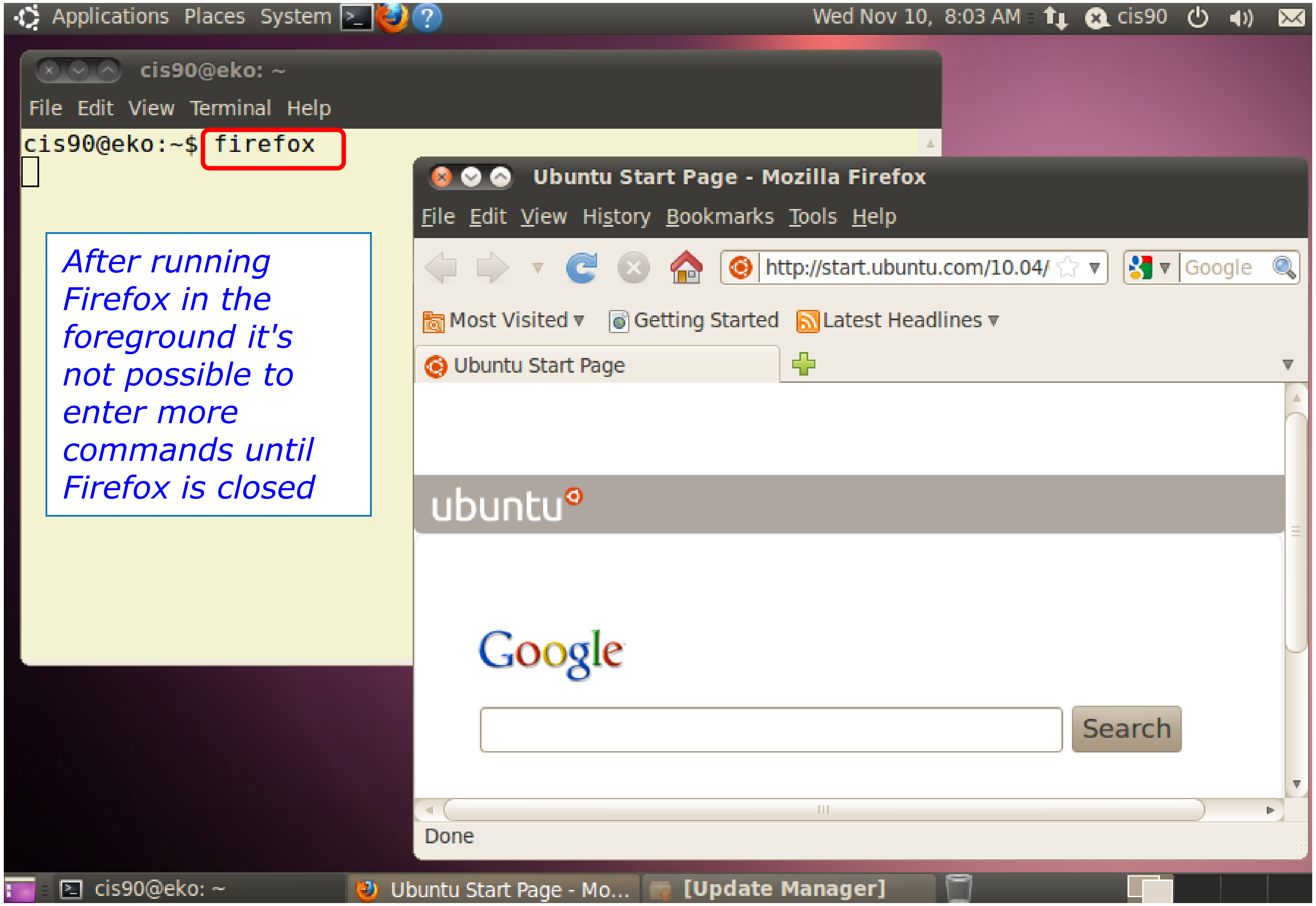
5) Write the signals that kill **annoy** into the chat window

Using &

to run a command in the background

Job Control

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



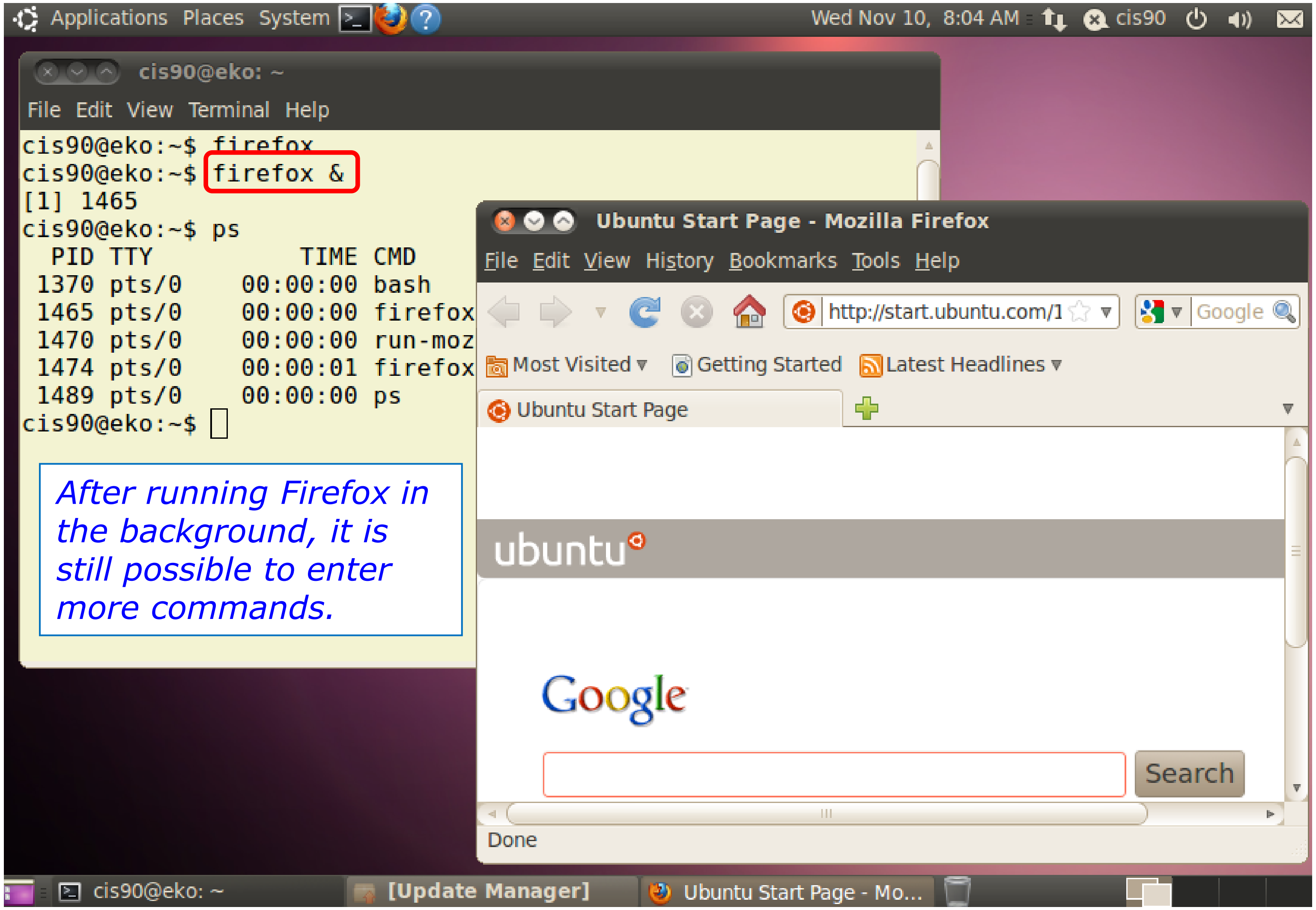
```
cis90@eko:~$ firefox
```

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

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]

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/simmsben $ find / -user 1200 2> duh | sort > huh
```

No prompt

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

Example 2

```
/home/cis90/simmsben $ find / -user 1200 2> duh | sort > huh &
[1] 11601
/home/cis90/simmsben $ date
Tue Nov  9 14:38:35 PST 2010
```

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

Job Control

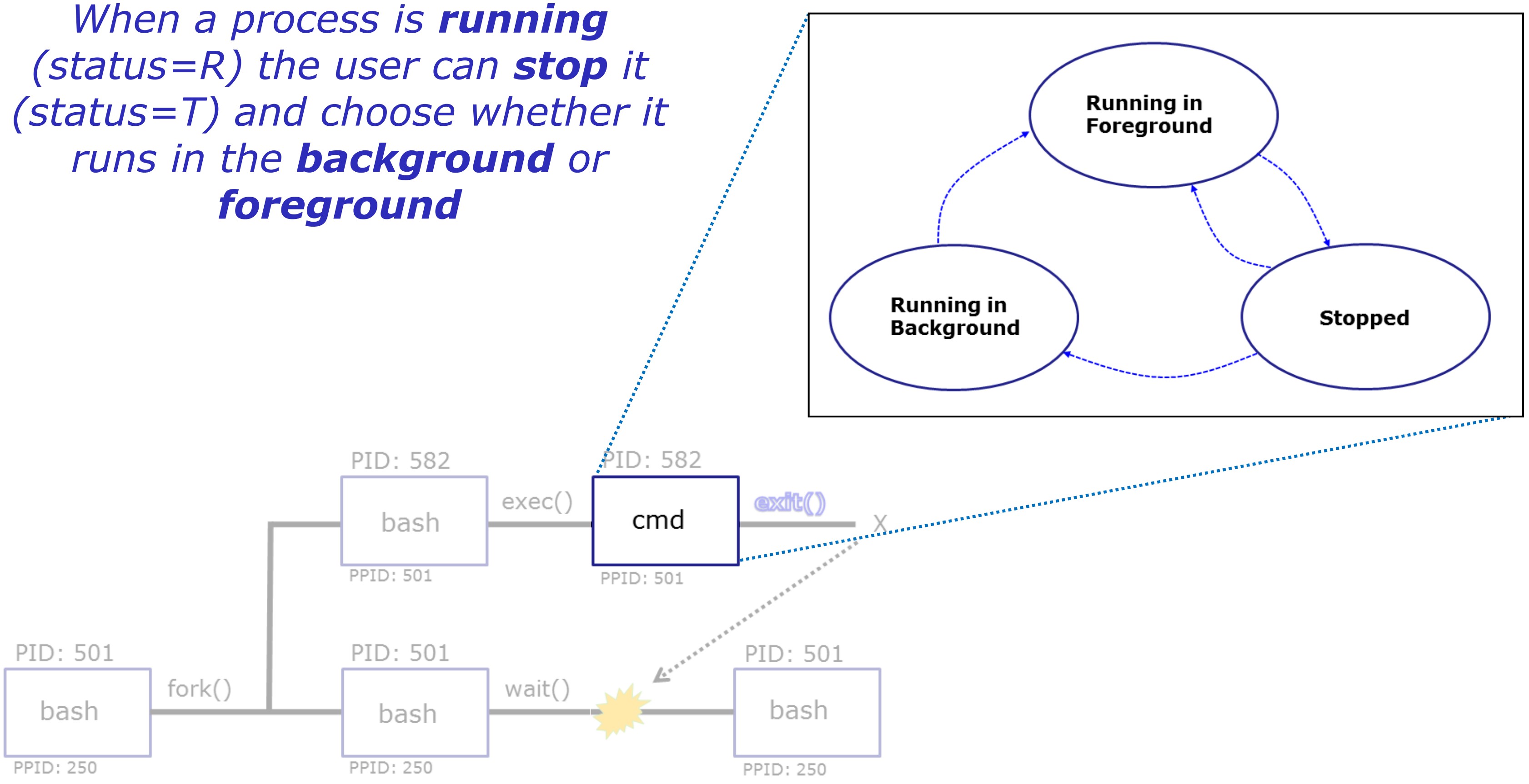
Job Control
A feature of the bash shell

&	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

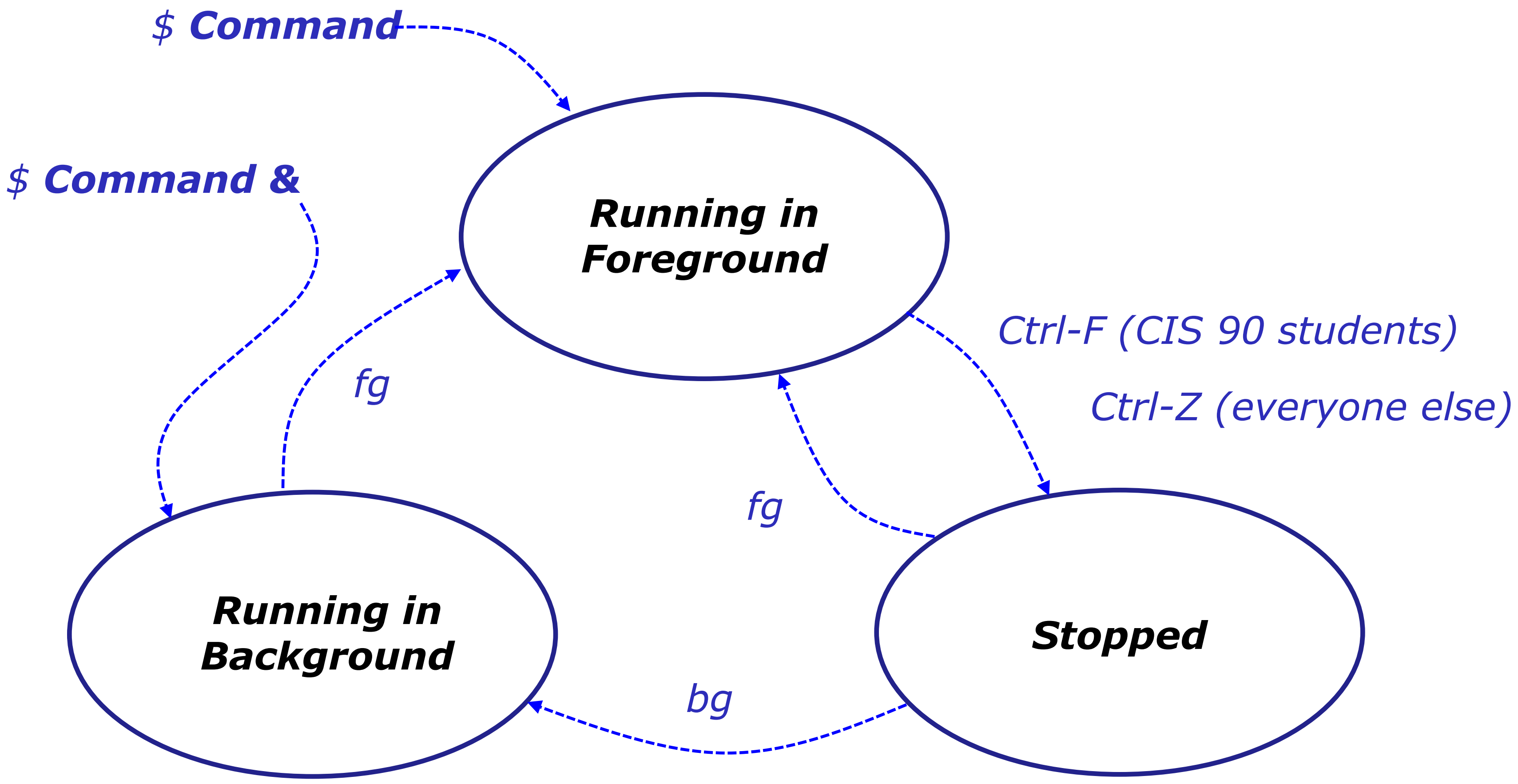
*Use **jobs**, **bg**, **fg** to list and resume jobs in the foreground or background*

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

How is yours configured?

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!

Review of Load Balancing

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 5/5/2012

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:

```
[rsimms@oslab ~]$ tty
/dev/pts/4

[rsimms@oslab ~]$ at now+2 minutes
at> echo "Take Benji for a walk" | mail -s "walk the dog" $LOGNAME
at> echo "Read your mail" > /dev/pts/4
at> <EOT>
job 11 at 2012-11-05 11:02
[rsimms@oslab ~]$ atq
11      2012-11-05 11:02 a rsimms
[rsimms@oslab ~]$
```

These should match

Type what happens in the chat window:

text editors

There are lots of text editors …

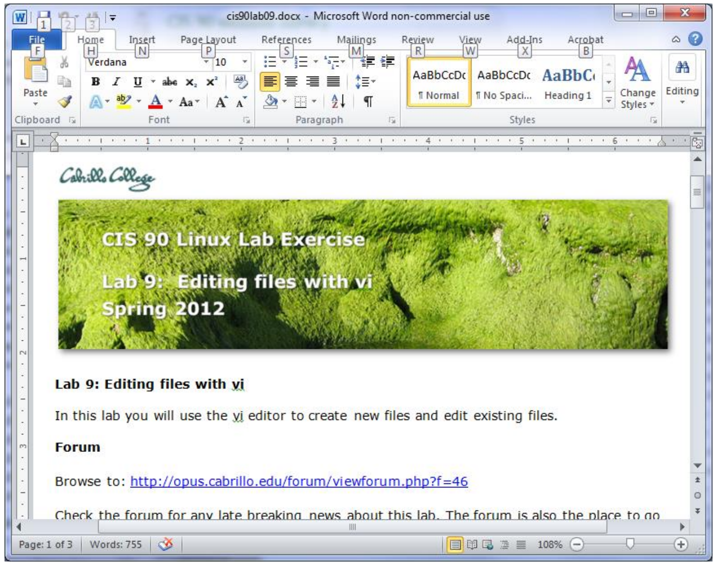
Windows
- notepad
- notepad++
- textpad

Mac
- TextWrangler

Linux
- gedit
- emacs
- nano
- vi

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 Linux Lab Exercise

Lab 9: Editing files with vi
Spring 2012

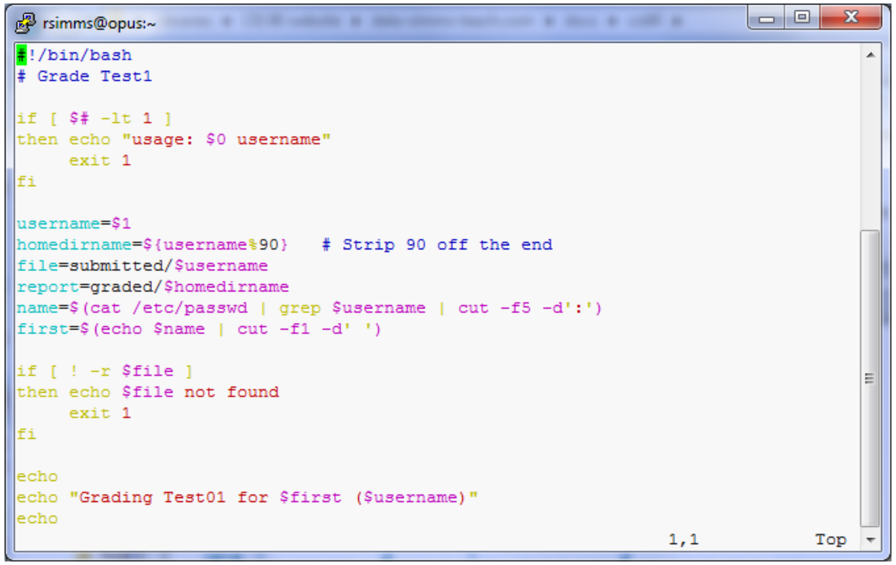
Lab 9: Editing files with vi

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 we are actually running VIM

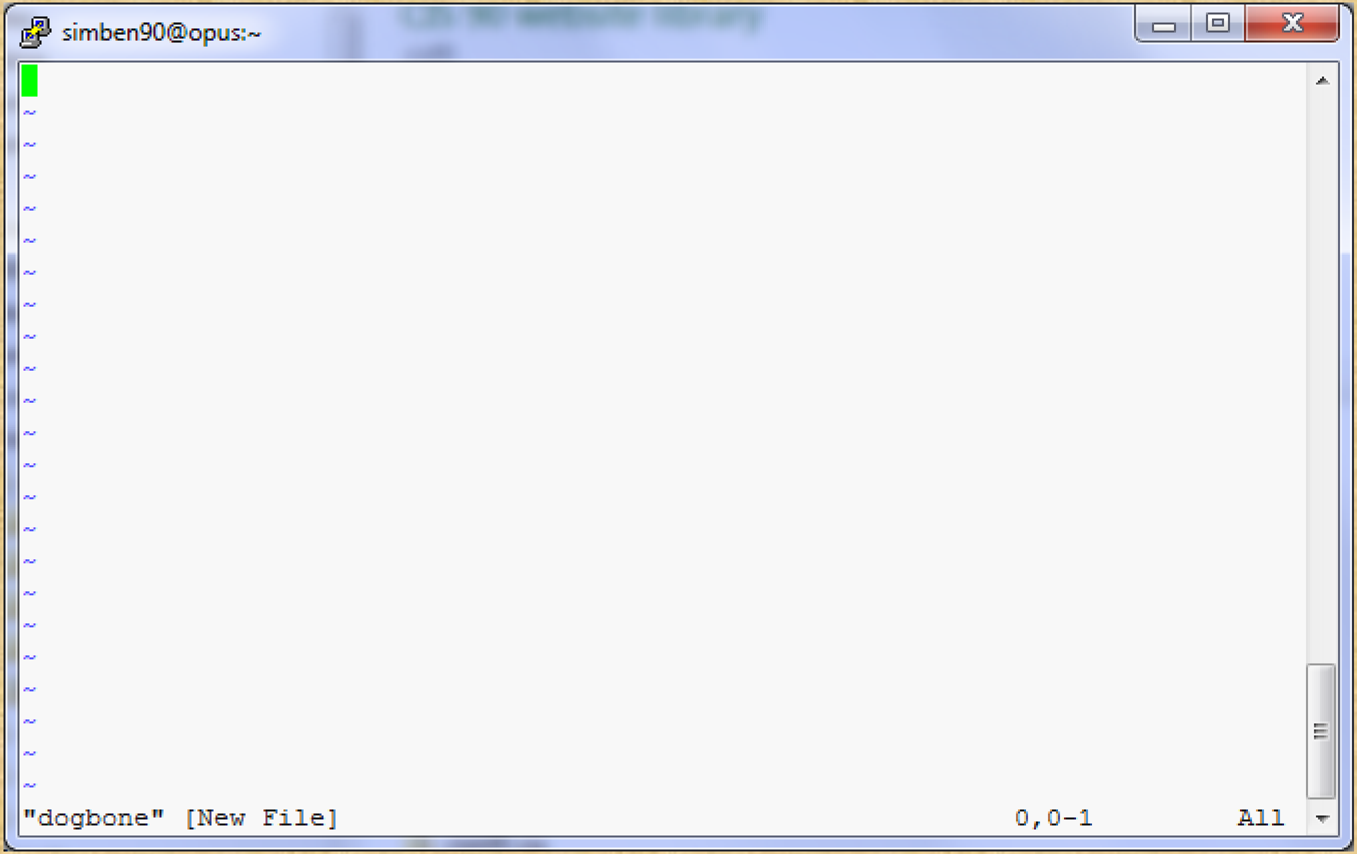
```
/home/cis90/simben $ type -a vi
vi is aliased to `vim'
vi is /bin/vi
/home/cis90/simben $ type vim
vim is hashed (/usr/bin/vim)
```

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

Type this

See this …

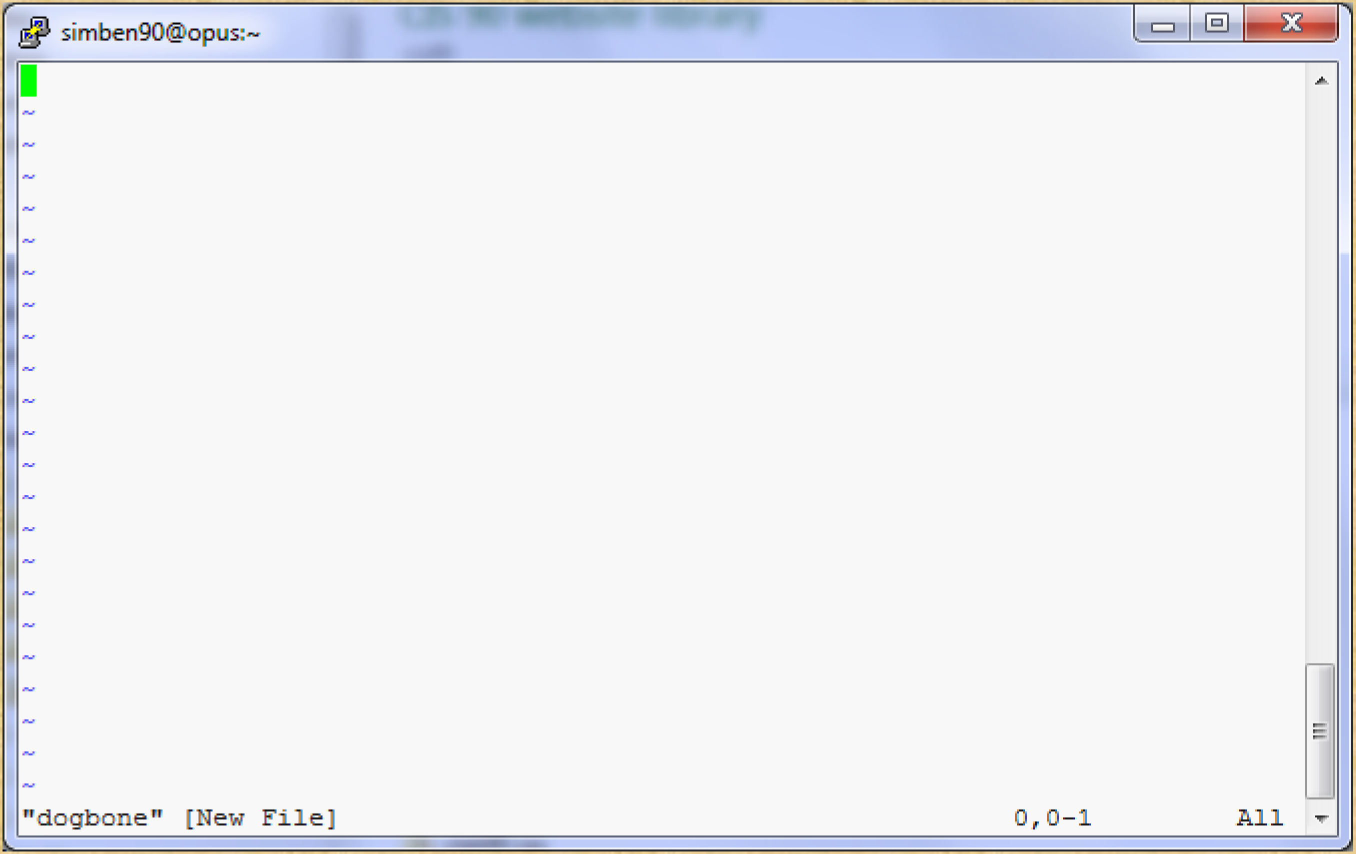


simben90@opus:~

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

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

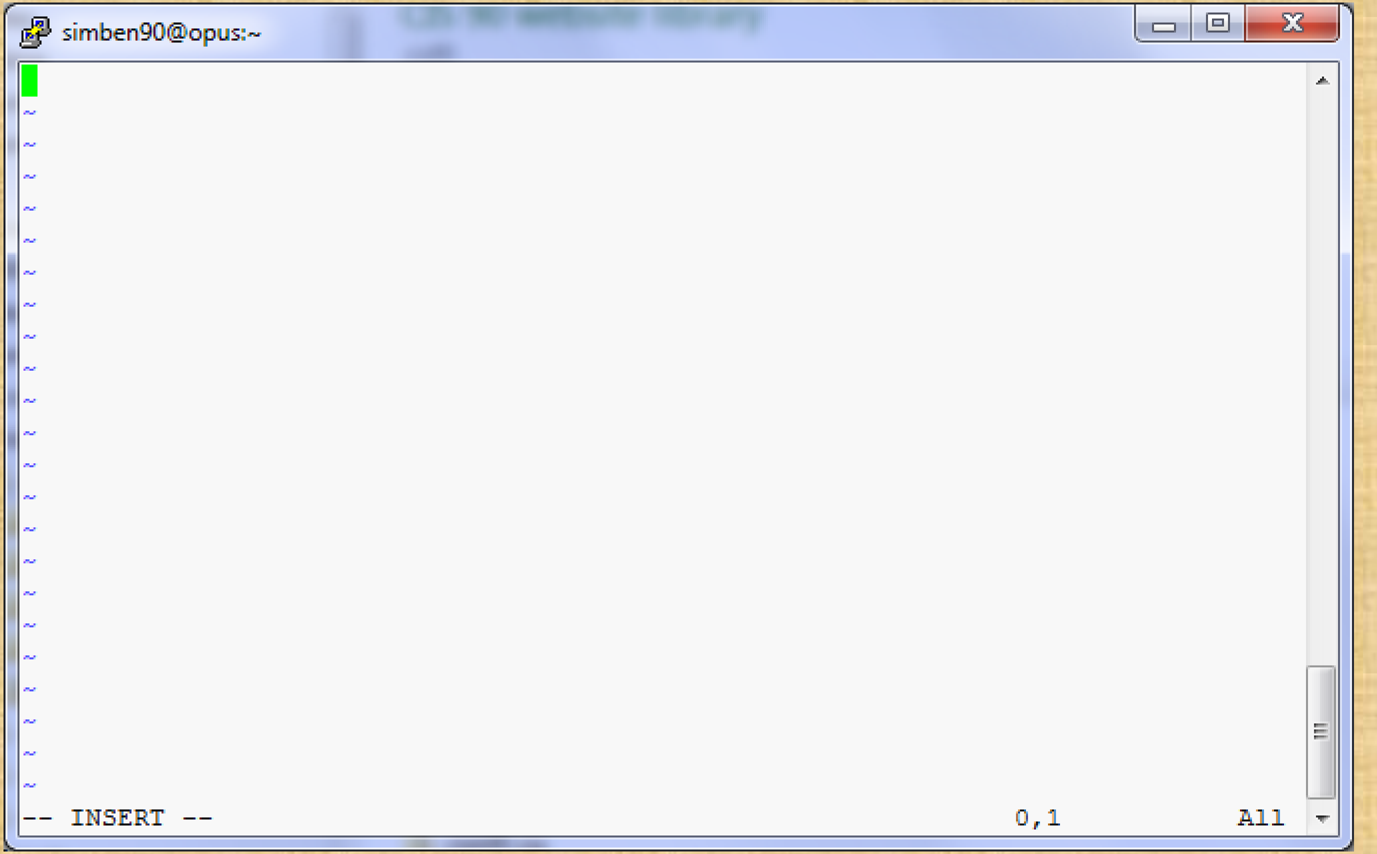
Tap the letter ***i*** *key (for insert)*



```
simben90@opus:~

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

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

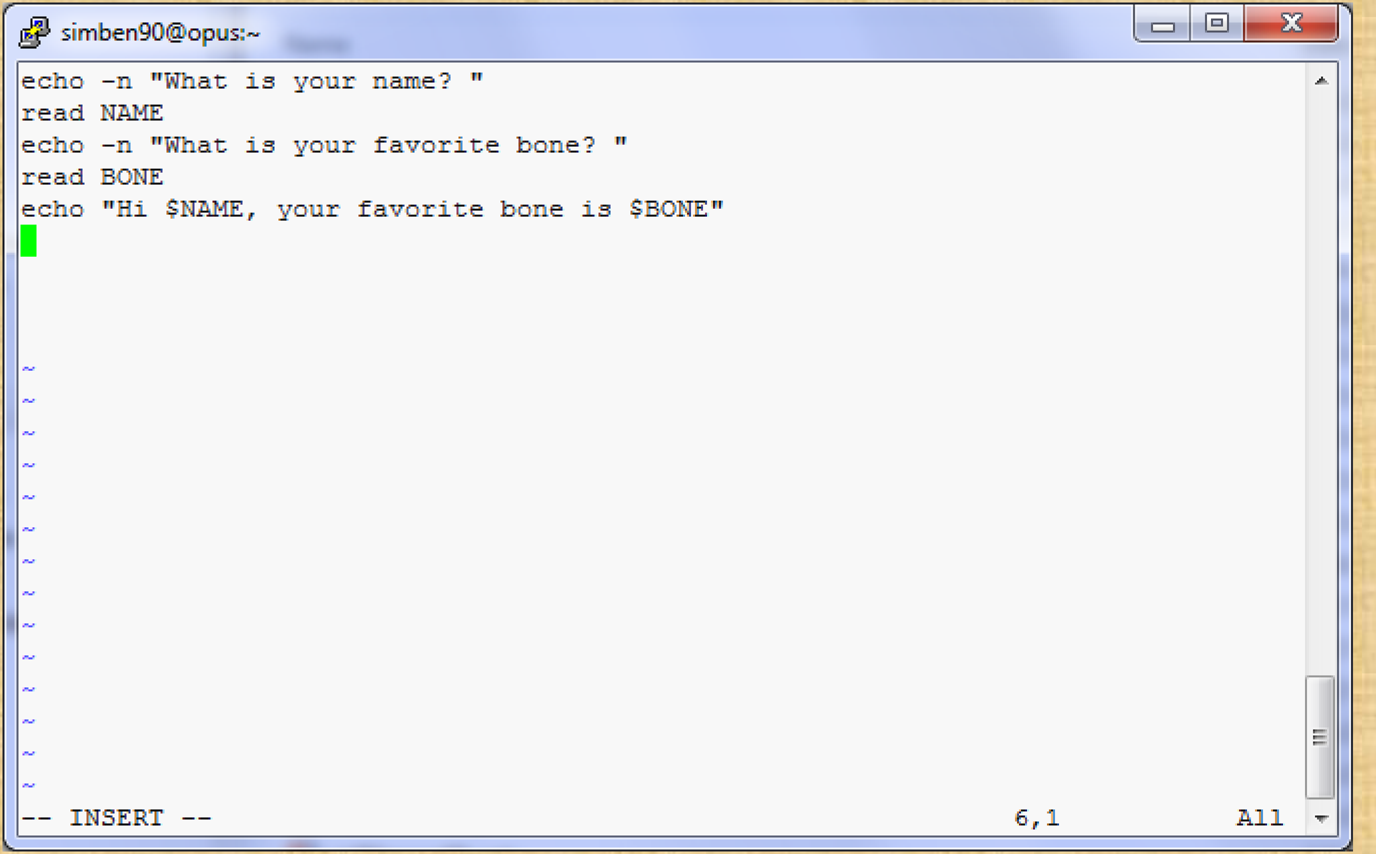
See this …



```
simben90@opus:~

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

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

Very carefully type these five lines

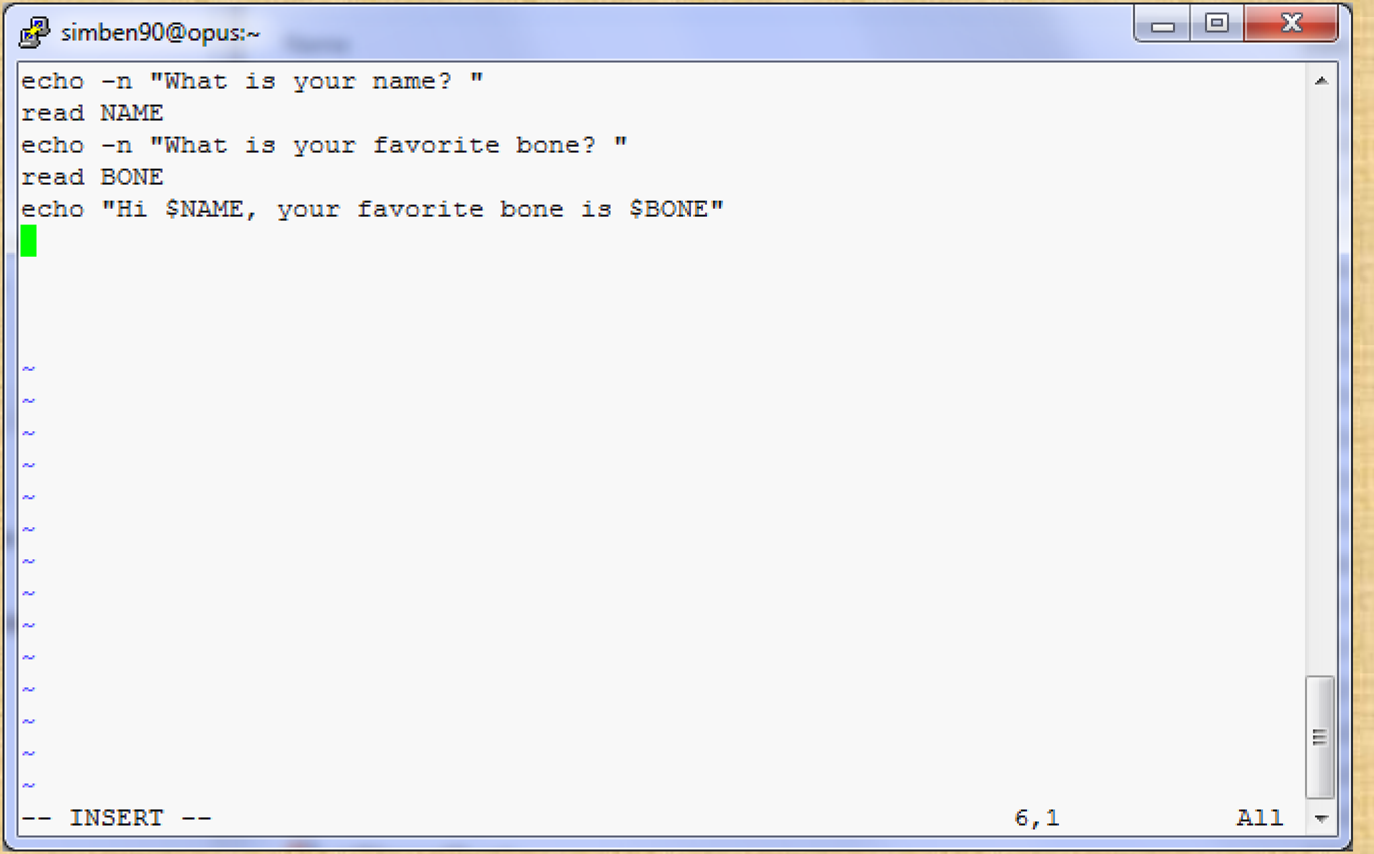


simben90@opus:~

```
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"

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

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

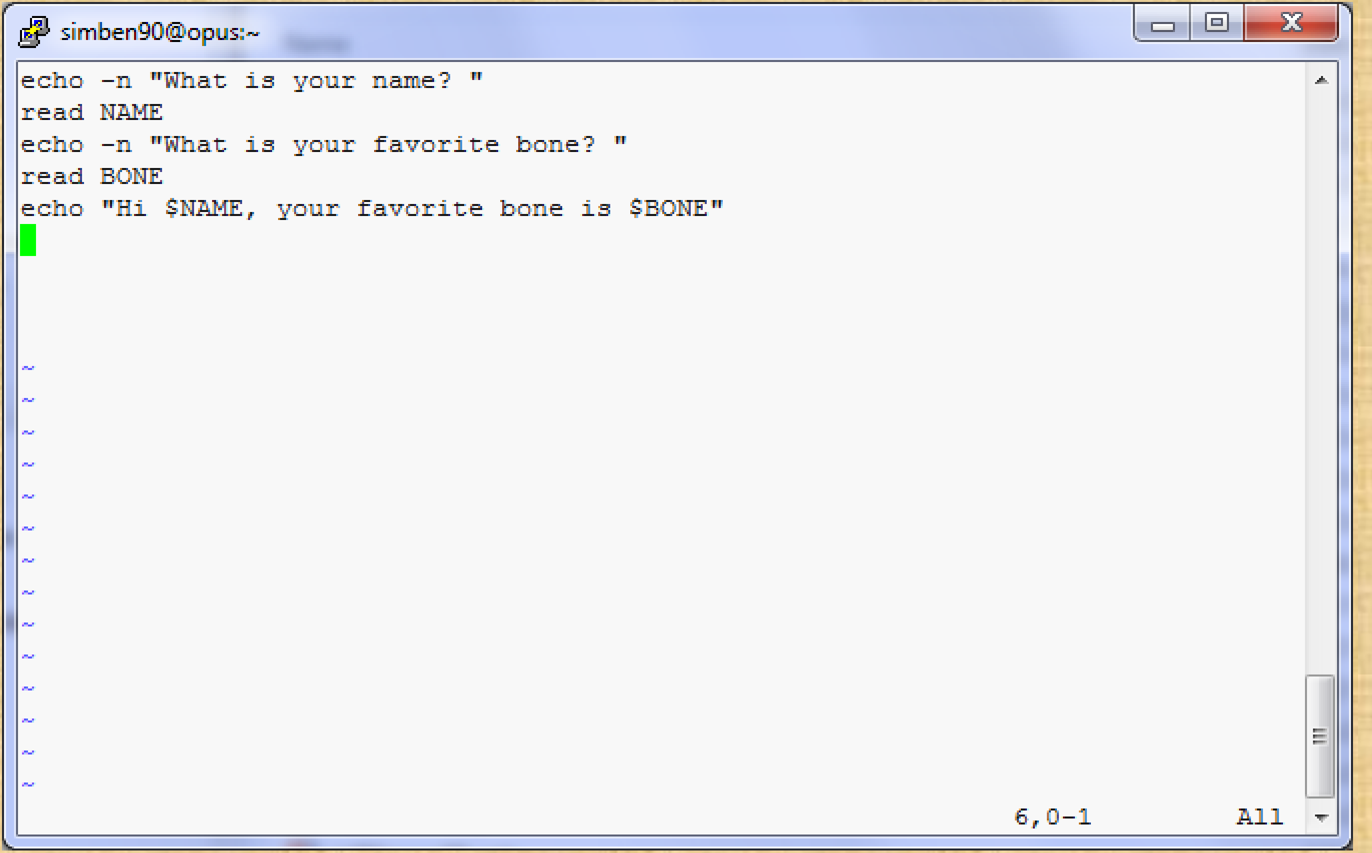
Have your neighbor check that your five lines are <u>PERFECT</u>



```
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
```

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

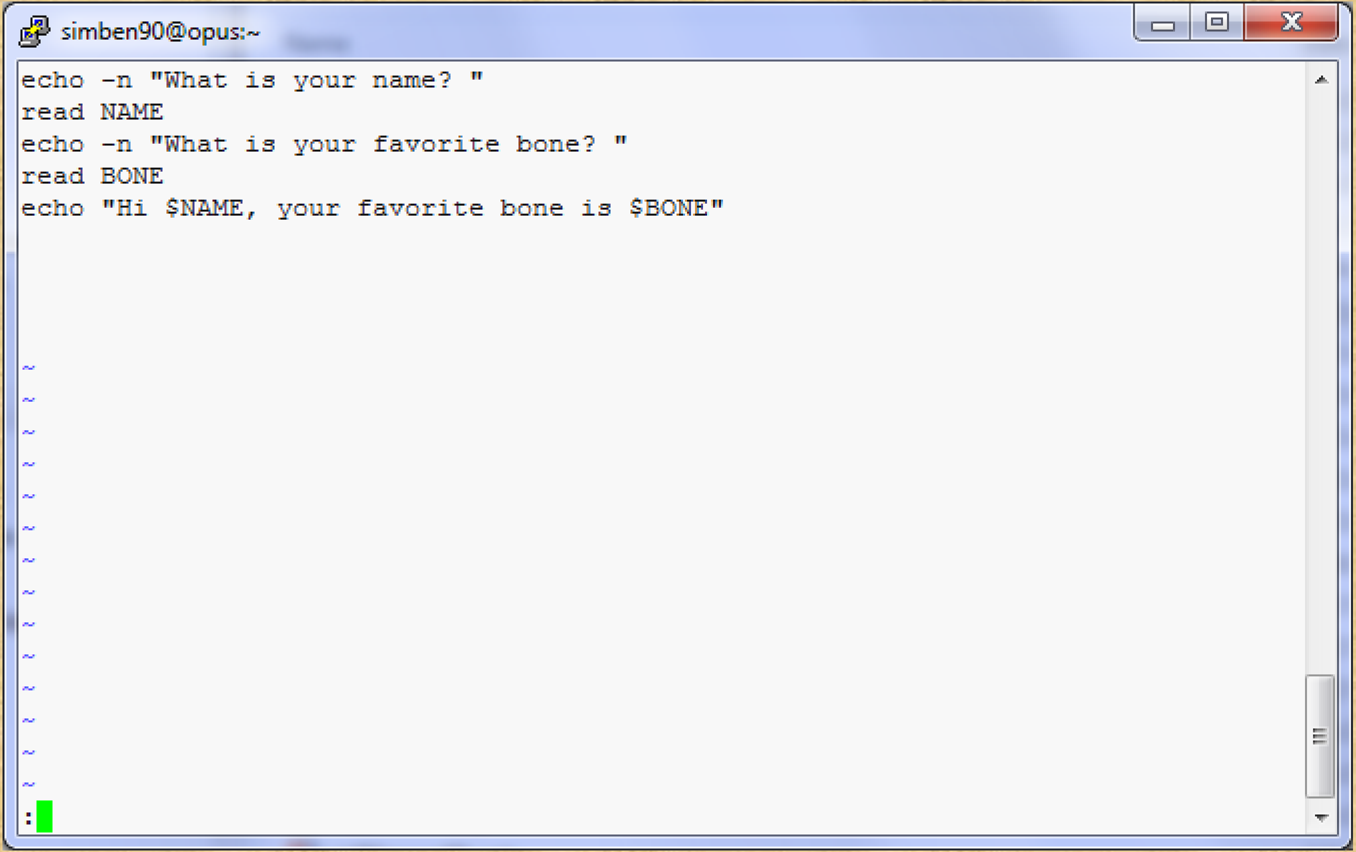
*Tap the **esc** key*



```
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
```

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

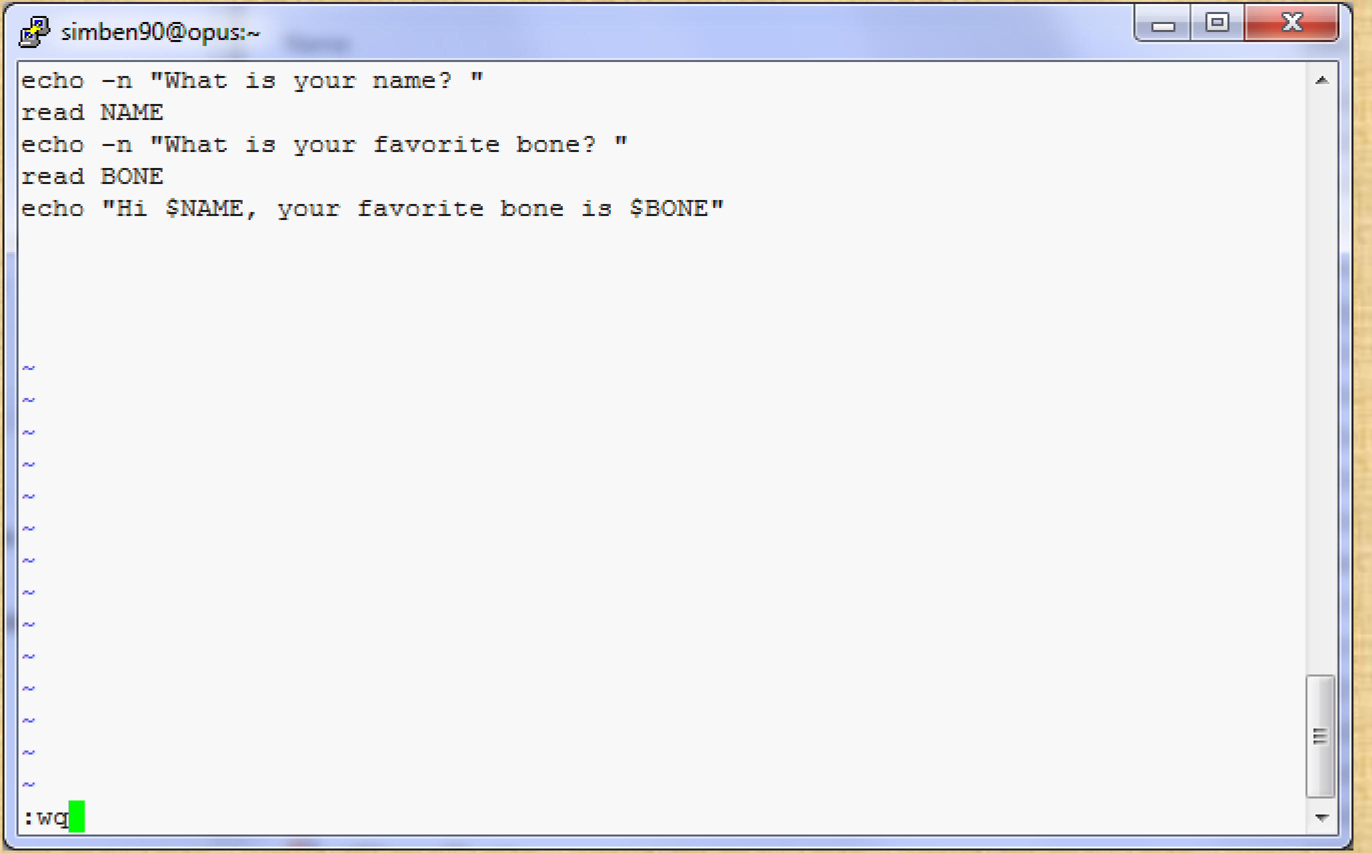
Type a **:**



```
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"








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

Keep 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
```

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

Tap the enter key

```
/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

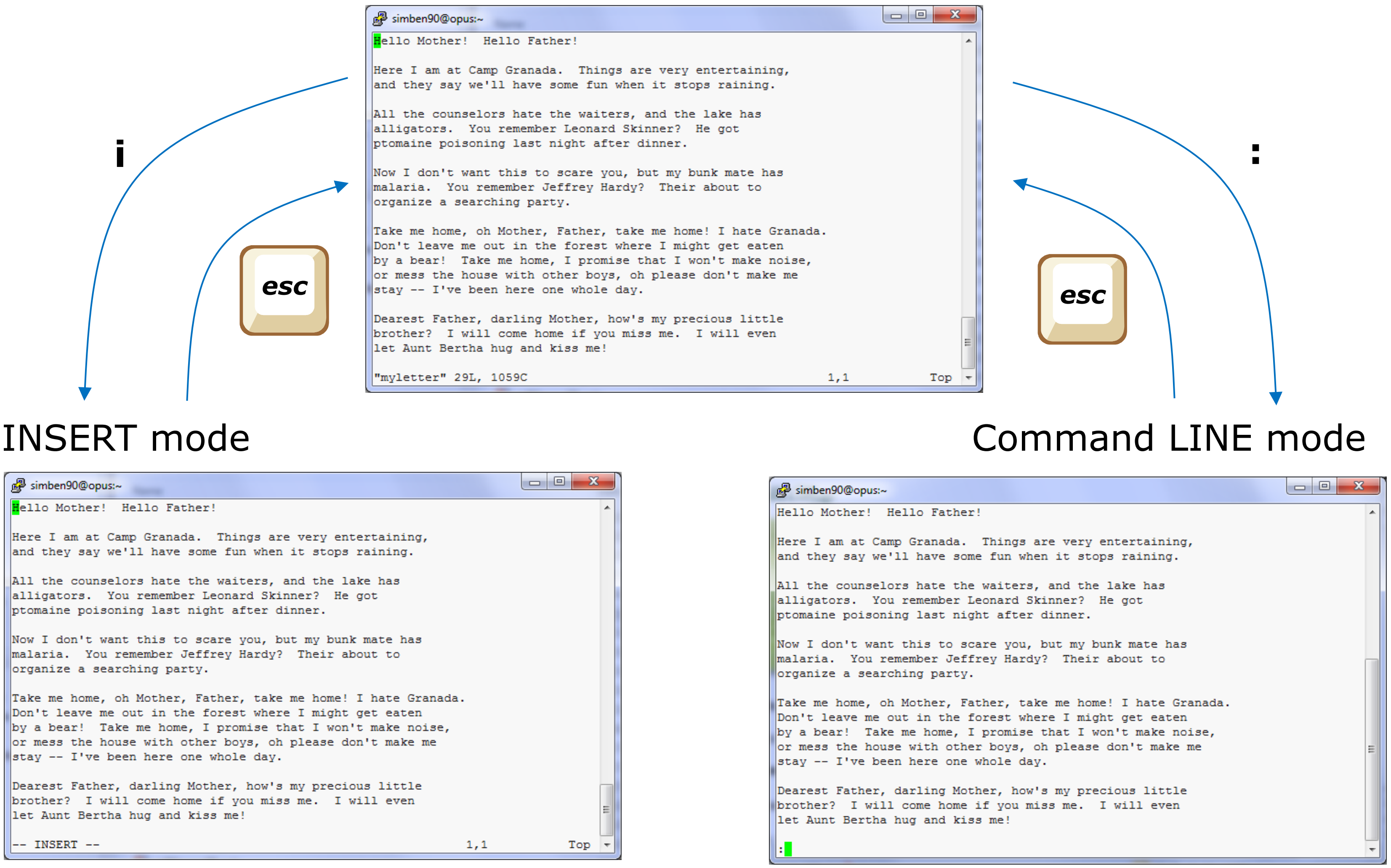
CIS 90 - Lesson 11

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

COMMAND mode

INSERT mode

Command LINE mode



```
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!

"myletter" 29L, 1059C                                    1,1           Top
```

```
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 --                                             1,1           Top
```

```
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!

:
```

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

vi
Moving around in a file

Use in COMMAND mode

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

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

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 Quiting

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*

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

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

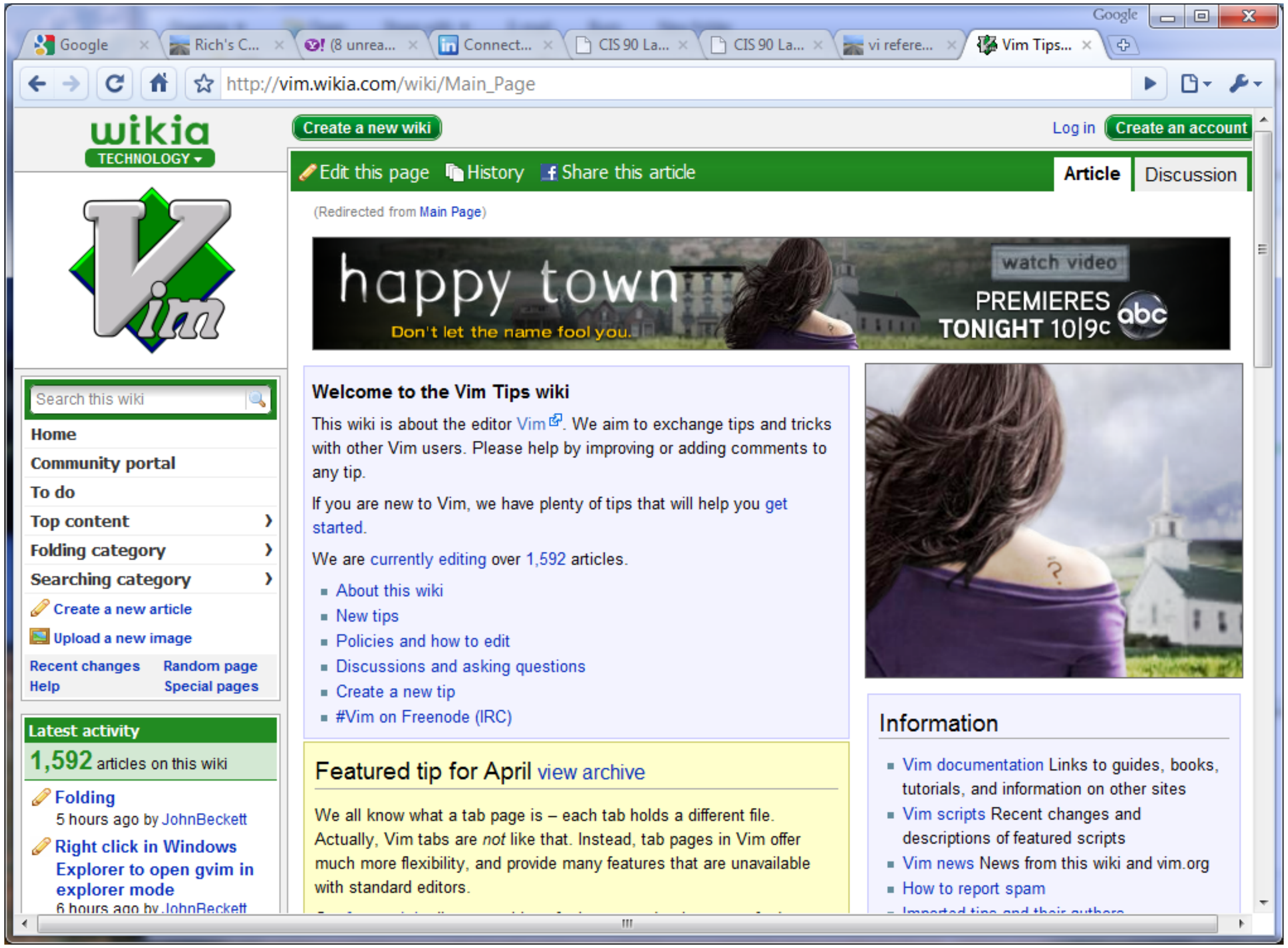
Use vi to edit your *edits/text.err* file

```
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.

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

Latest activity

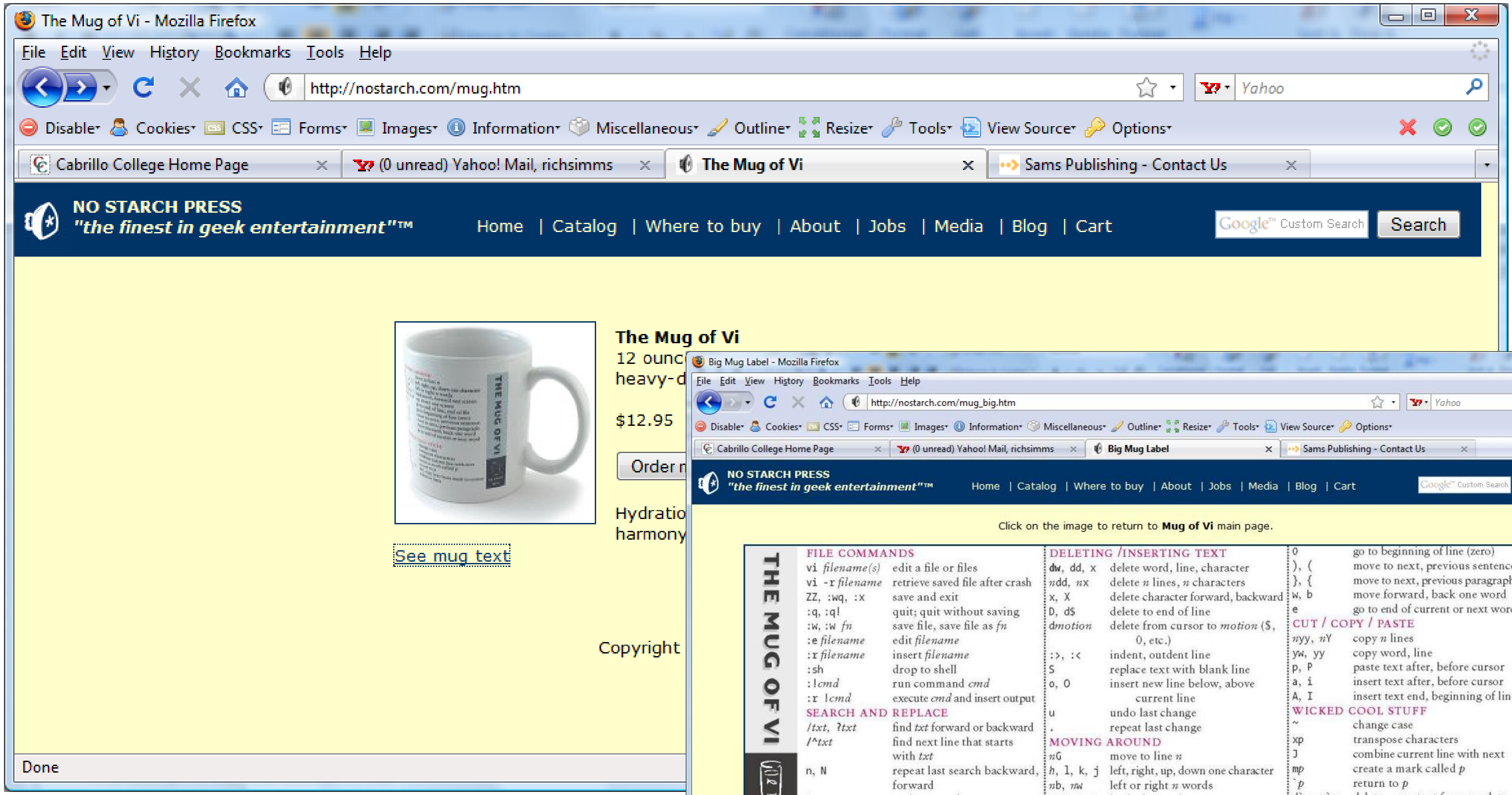
1,592 articles on this wiki

Folding
5 hours ago by JohnBeckett

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

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
Yahoo
Disable Cookies CSS Forms Images Information Miscellaneous Outline Resize Tools View Source Options
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
Google Custom Search
Search
Done

The Mug of Vi
12 ounc
heavy-d

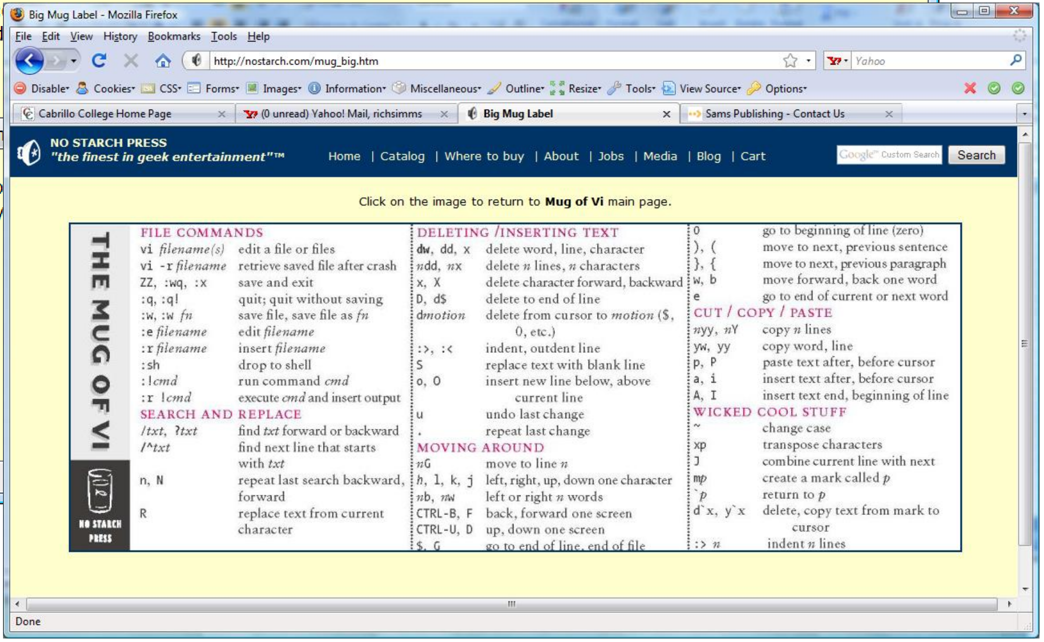
$12.95

Order n

Hydratio
harmony

See mug text

Copyright


Big Mug Label - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://nostarch.com/mug_big.htm
Yahoo
Disable Cookies CSS Forms Images Information Miscellaneous Outline Resize Tools View Source Options
Cabrillo College Home Page
(0 unread) Yahoo! Mail, richsimms
Big Mug Label
Sams Publishing - Contact Us
NO STARCH PRESS
"the finest in geek entertainment"™
Home | Catalog | Where to buy | About | Jobs | Media | Blog | Cart
Google Custom Search
Search
Done

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

THE MUG OF VI

NO STARCH PRESS

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

http://nostarch.com/mug.htm

/bin/mail and vi

```
/home/cis90/simmsben $ mail roddyduk
Subject: Good bones
Hey Duke,
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
```

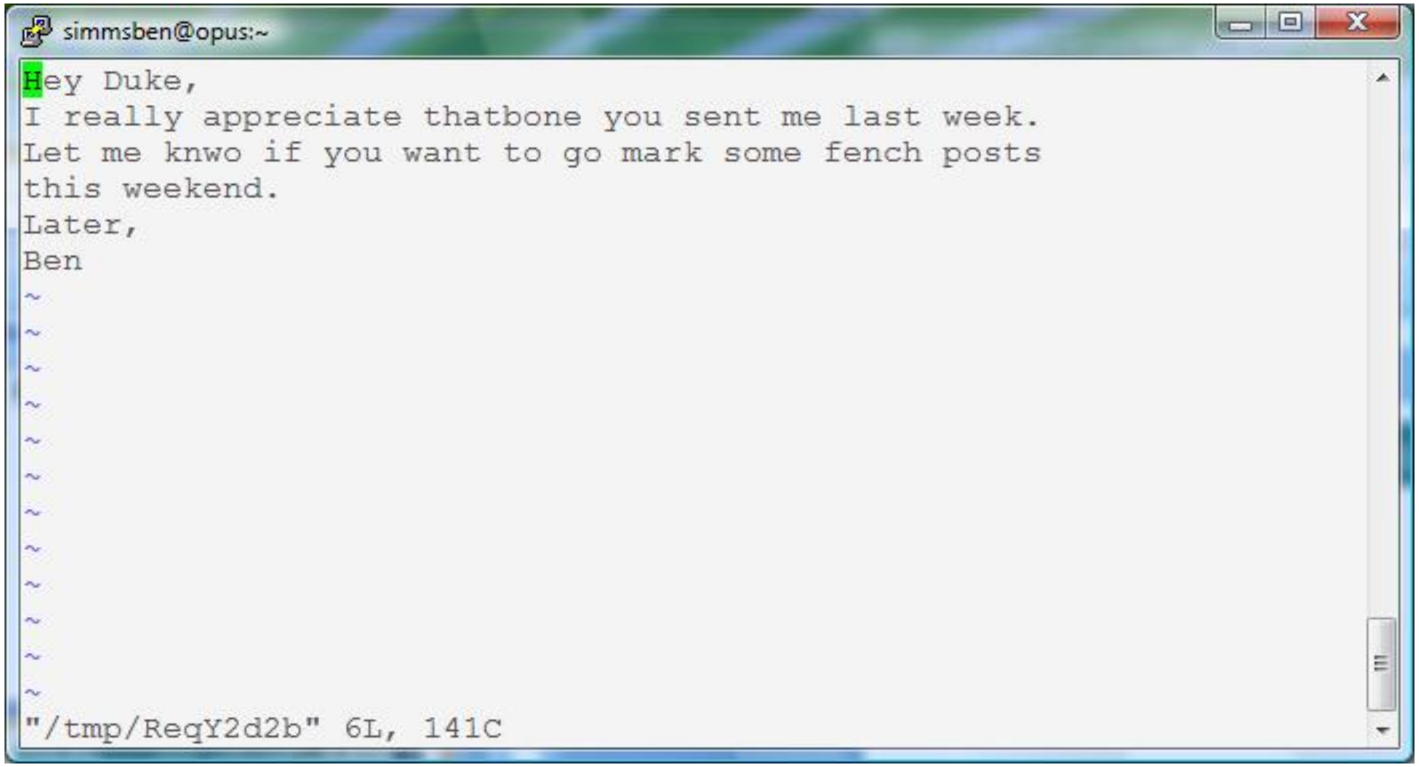
You are composing a message and you spot some typos … CRUD … what can you do?

/bin/mail and vi

```
/home/cis90/simmsben $ mail roddyduk
Subject: Good bones
Hey Duke,
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



```
Hey Duke,
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
~
~
~
~
~
~
~
~
~
~
~
~
"/tmp/ReqY2d2b" 6L, 141C
```

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/simmsben $ mail roddyduk
Subject: Good bones
Hey Duke,
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)
.
Cc:
/home/cis90/simmsben $
```

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

/bin/mail and vi

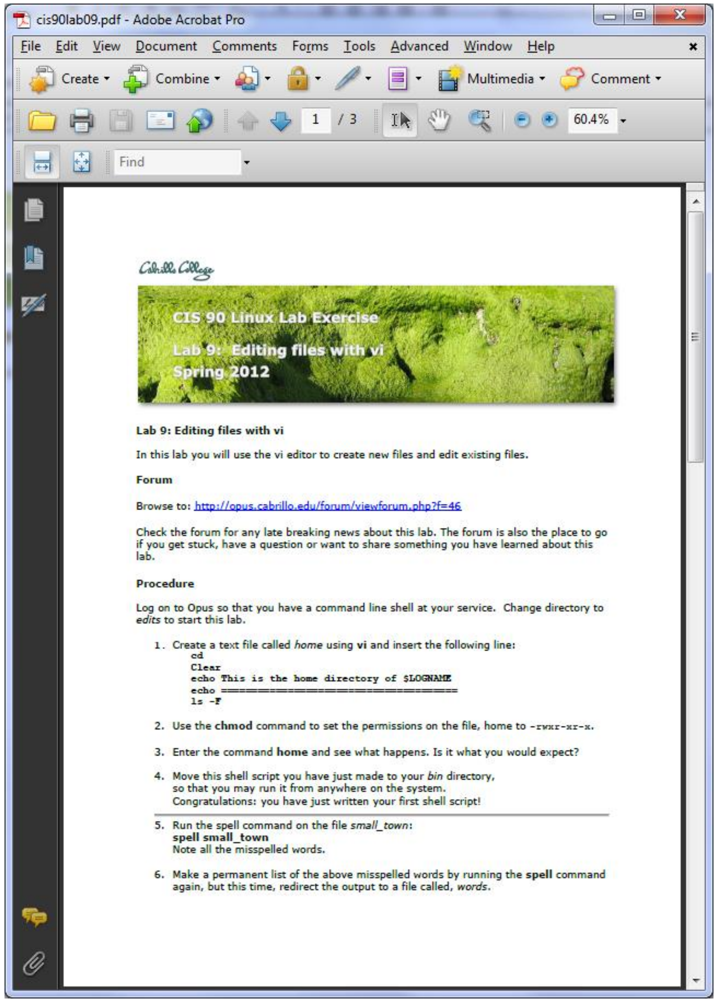
```
/home/cis90/roddyduk $ mail
Mail version 8.1 6/6/93.  Type ? for help.
"/var/spool/mail/roddyduk": 1 message 1 unread
>U  1 simmsben@opus.cabril  Mon Nov 10 20:25  22/782   "Good bones"
& 1
Message 1:
From simmsben@opus.cabrillo.edu  Mon Nov 10 20:25:32 2008
Date: Mon, 10 Nov 2008 20:25:32 -0800
From: Benji Simms <simmsben@opus.cabrillo.edu>
To: roddyduk@opus.cabrillo.edu
Subject: Good bones

Hey Duke,
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,
Ben

&
```

The message Duke 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 $
```

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


cis90lab09.pdf - Adobe Acrobat Pro
File Edit View Document Comments Forms Tools Advanced Window Help
Create Combine Multimedia Comment
1 / 3
60.4%
Find

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 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 line:
   ```
   cd
   Clear
   echo This is the home directory of $LOGNAME
   echo ===================================
   ls -F
   ```
2. Use the **chmod** command to set the permissions on the file, home to `-rwxr-xr-x`.
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!

Instructor: remember to mail students the tech file!

~/cis90/lab09/mail-tech-all

A Tangent on Spell

spell command

```
/home/cis90/roddyduk/edits $ cat text
Welcome to the CIS 90 class !!
```

```
/home/cis90/roddyduk/edits $ spell text
CIS
```

***spell** command flags CIS as misspelled word.*

How can we add CIS to the dictionary?

spell command

```
/home/cis90/roddyduk/edits $ cat text
Welcome to the CIS 90 class !!
/home/cis90/roddyduk/edits $ spell text
CIS
```

How can we add CIS to the dictionary?

```
/home/cis90/roddyduk/edits $ man spell
No manual entry for spell
/home/cis90/roddyduk/edits $ type spell
spell is hashed (/usr/bin/spell)
/home/cis90/roddyduk/edits $ file usr/bin/spell
/usr/bin/spell: Bourne shell script text executable
/home/cis90/roddyduk/edits $ cat /usr/bin/spell
#!/bin/sh

# aspell list mimicks the standard unix spell program, roughly.

cat "$@" | aspell list --mode=none | sort -u

/home/cis90/roddyduk/edits $
```

Hmmm. No man page for spell ???????????

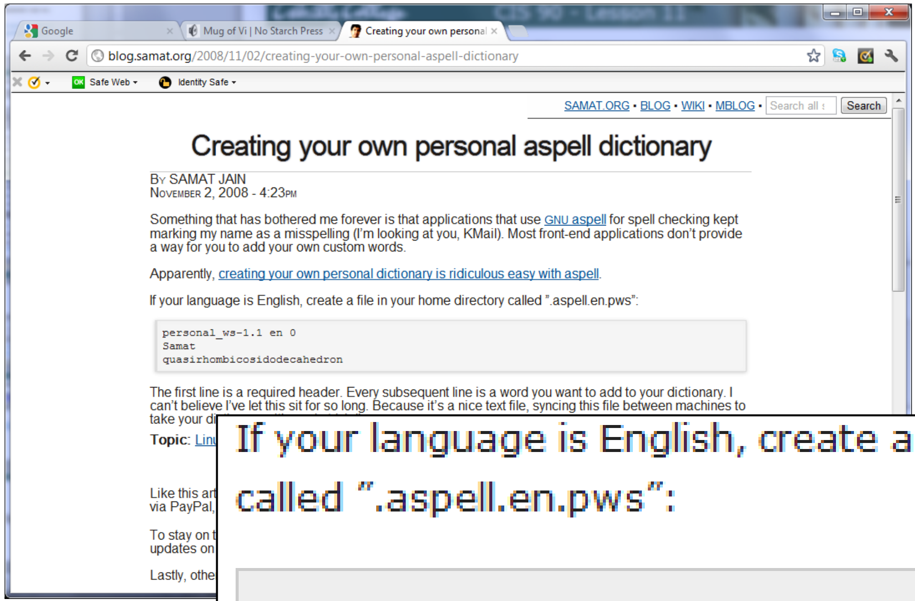
*OK, the actual command is **aspell***

spell command

```
ASPELL(1)              Aspell Abbreviated User’s Manual              ASPELL(1)

NAME
       aspell - interactive spell checker

SYNOPSIS
       aspell [options] <command>

DESCRIPTION
       aspell  is  a utility that can function as an ispell -a replacement,
       as an independent spell checker, as  a  test  utility  to  test  out
       Aspell features, and as a utility for managing dictionaries.

COMMANDS
       <command> is one of:

       -?,help
              display the help message

       -c,check file
              to spell-check a file
```

There must be a way to add CIS …. but … lets try google

spell command



Google | Mug of Vi | No Starch Press | Creating your own personal

blog.samat.org/2008/11/02/creating-your-own-personal-aspell-dictionary

SAMAT.ORG • BLOG • WIKI • MBLOG • Search

Creating your own personal aspell dictionary

By SAMAT JAIN
November 2, 2008 - 4:23pm

Something that has bothered me forever is that applications that use GNU aspell for spell checking kept marking my name as a misspelling (I'm looking at you, KMail). Most front-end applications don't provide a way for you to add your own custom words.

Apparently, creating your own personal dictionary is ridiculous easy with aspell.

If your language is English, create a file in your home directory called ".aspell.en.pws":

```
personal_ws-1.1 en 0
Samat
quasirhombicosidodecahedron
```

The first line is a required header. Every subsequent line is a word you want to add to your dictionary. I can't believe I've let this sit for so long. Because it's a nice text file, syncing this file between machines to take your di

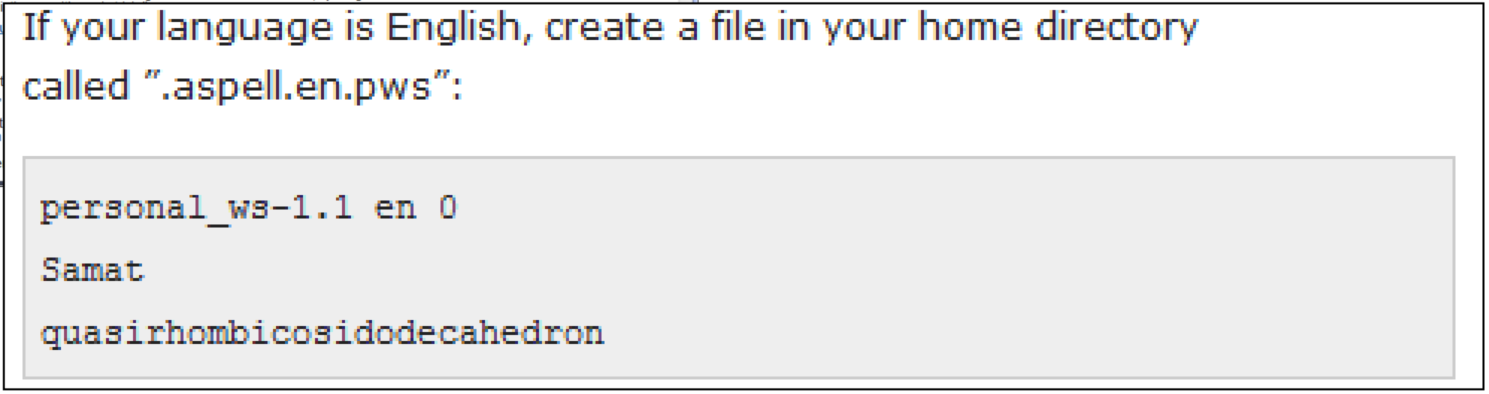
Topic: Linu

Like this ar
via PayPal,

To stay on t
updates on

Lastly, othe

How to add words to your dictionary



If your language is English, create a file in your home directory called ".aspell.en.pws":

```
personal_ws-1.1 en 0
Samat
quasirhombicosidodecahedron
```

Googling "linux aspell personal dictionary" yields this page

Bingo! Thank you Samat Jain

spell command

```
/home/cis90/roddyduk/edits $ cd
/home/cis90/roddyduk $ echo "personal_ws-1.1 en 0" >  .aspell.en.pws
/home/cis90/roddyduk $ echo "CIS" >> .aspell.en.pws
/home/cis90/roddyduk $ cd edits/
/home/cis90/roddyduk/edits $ spell text
```

This is how you would add your own custom dictionary to be used with spell checks

```
/home/cis90/simben $ cat edits/spellk
Spell Check


Eye halve a spelling chequer
It came with my pea sea
It plainly marques four my revue
Miss steaks eye kin knot sea.
Eye strike a key and type a word
And weight four it two say
Weather eye am wrong oar write
It shows me strait a weigh.
As soon as a mist ache is maid
It nose bee fore two long
And eye can put the error rite
Its rare lea ever wrong.
Eye have run this poem threw it
I am shore your pleased two no
Its letter perfect awl the weigh
My chequer tolled me sew.

/home/cis90/simben $ spell edits/spellk
chequer
```

How would you add "chequer" (the British spelling) to your personal dictionary?

Copy the commands used into the chat window when finished

```
$ echo "personal_ws-1.1 en 0" > ~/.aspell.en.pws
$ echo "chequer" >> ~/.aspell.en.pws
```

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 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

```
/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;
```

```
[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;
```

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

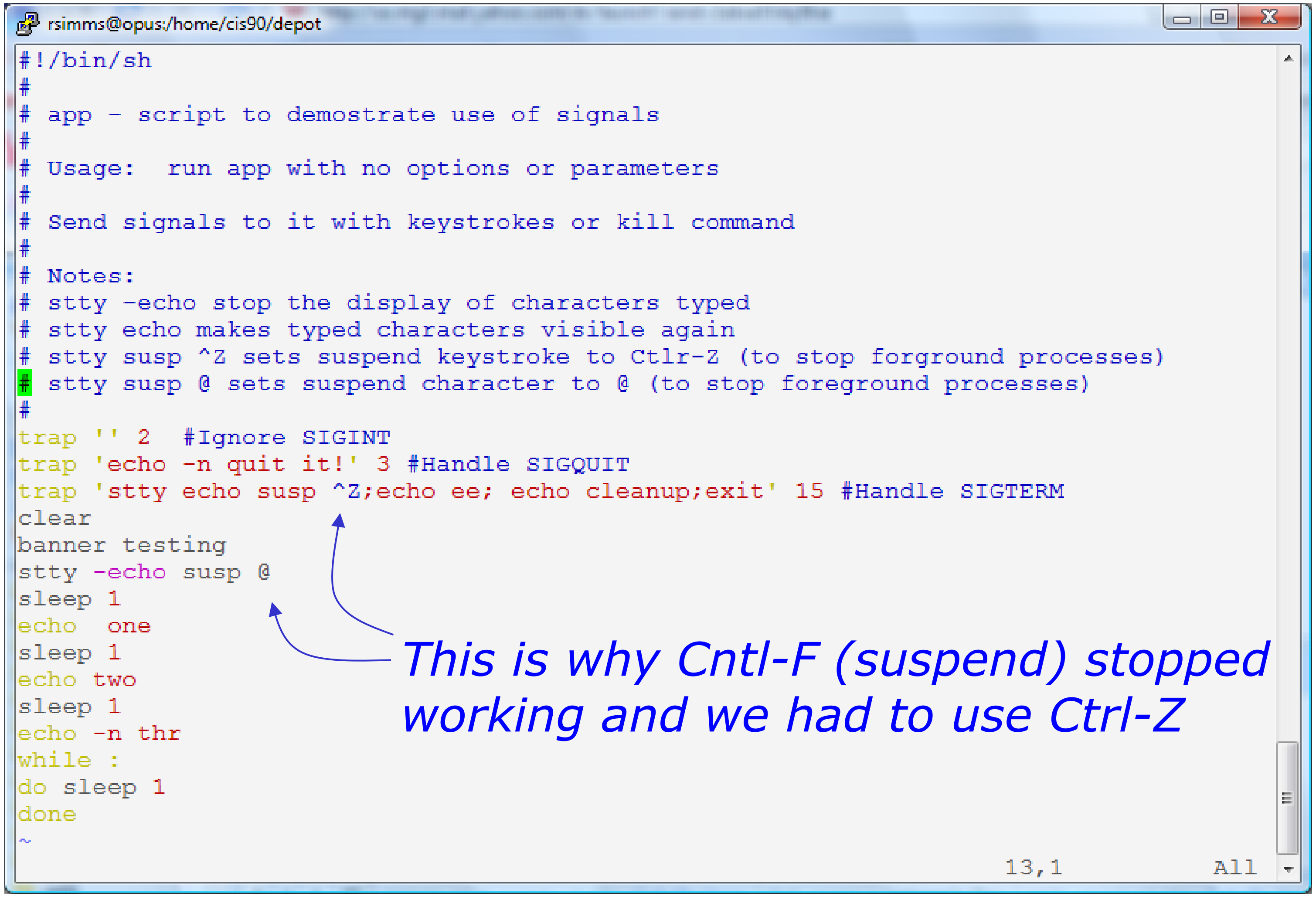
```
[rsimms@opus ~]$ sleep 5

[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 Cntl-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

stdin
stdout
stderr
18

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