Lesson Module Checklist
- Slides –
- Properties -
- Flash cards –
- First minute quiz –
- Web calendar summary –
- Web book pages –
- Commands –
- Lab 10 and Final Project –

- CCC Confer wall paper –
- riddle file copied to bin directory
- allscripts updated

- Materials uploaded –
- Backup headset charged –
- Backup slides, CCC info, handouts on flash drive -

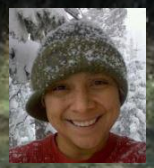
- Polycom
- Check that room headset is charged – done

CIS 90 - Lesson 12

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


Dieskau

Jonathan


Ana

David


Obie

Dave

Cole

Corey

Nancy


Ryan

Elia


Tasha

Darren


Scott


Devin

Everett

Juan

Raven

Rogan


Mike


Mook


Melissa

Cameron

Jose

Jeff


Matt

Kenneth

Ousmane

Ian


Solomon

Henry

Matthew

Mason

Chan

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

Quiz #9

Please answer these questions in the order shown:

email answers to: **`risimms@cabrillo.edu`**
(within the first few minutes of class)

[ ] Has the phone bridge been added?

[ ] Is recording on?

[ ] Does the phone bridge have the mike?

[ ] Share slides, putty x 3, and Chrome

[ ] Disable spelling on PowerPoint

The Shell Environment

Objectives	Agenda
• Be able to set, view and unset shell variables • Describe the difference between the set and env commands • Explain the importance of the export command. • Describe three actions that are handled by the .bash_profile file • Define user-defined aliases • Explain the . (dot) command and the exec command.	• Quiz • Housekeeping • Spell checking • vi and /bin/mail • Review pathnames • Final project prep • Variables • The shell environment • Aliases • .bash_profile • .bashrc

Previous material and assignment

1. Lab 9 due midnight tonight

2. Five posts due 11:59PM tonight

   Reminder: Only posts between in the CIS 90 ONLINE forum between 3/29 and 5/2 (inclusive) are counted.

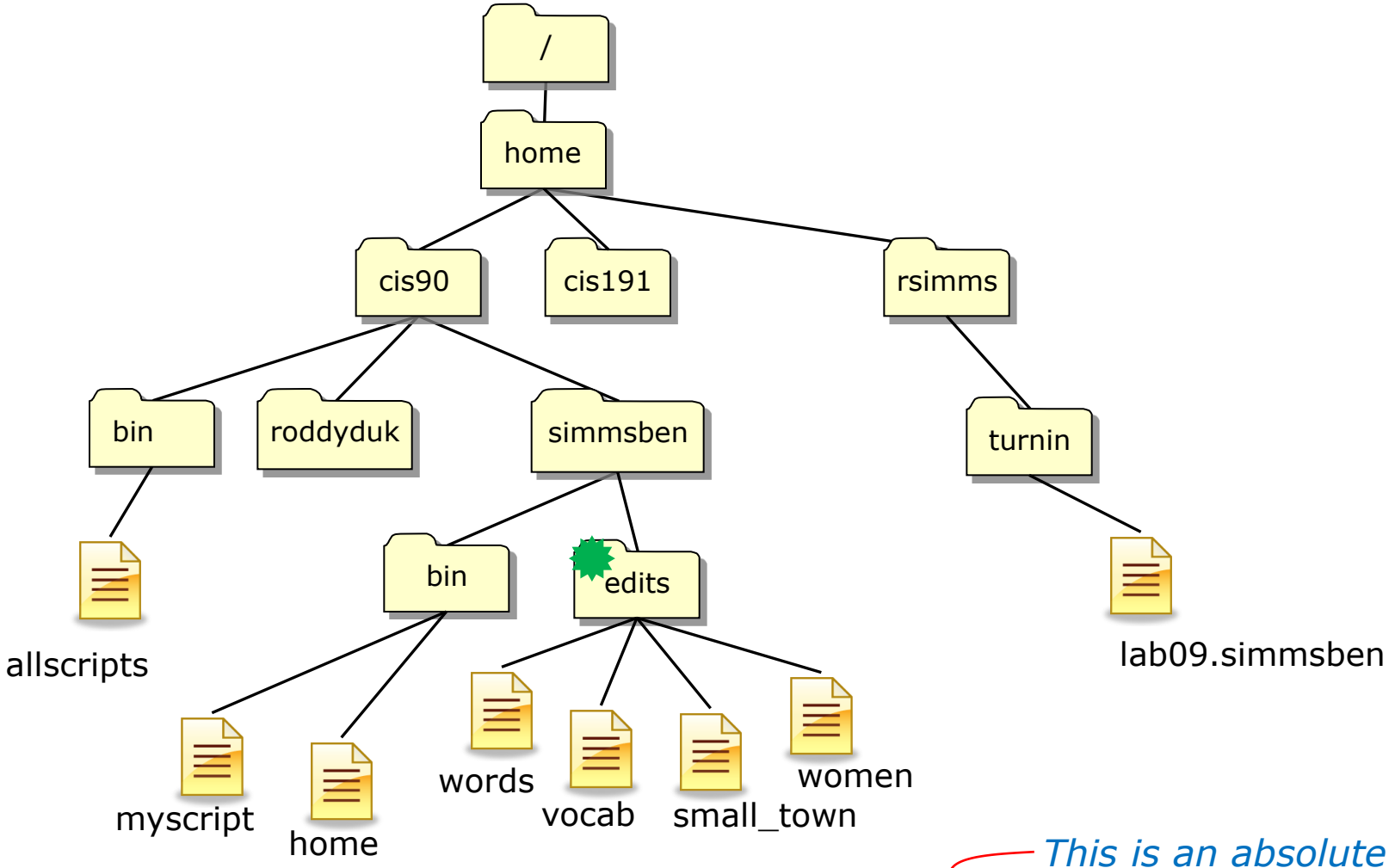
3. Questions?
   - vi
   - lab 9
   - previous material

pathnames

REMINDER

- You must **ALWAYS** use complete pathnames when specifying files as arguments on a command.

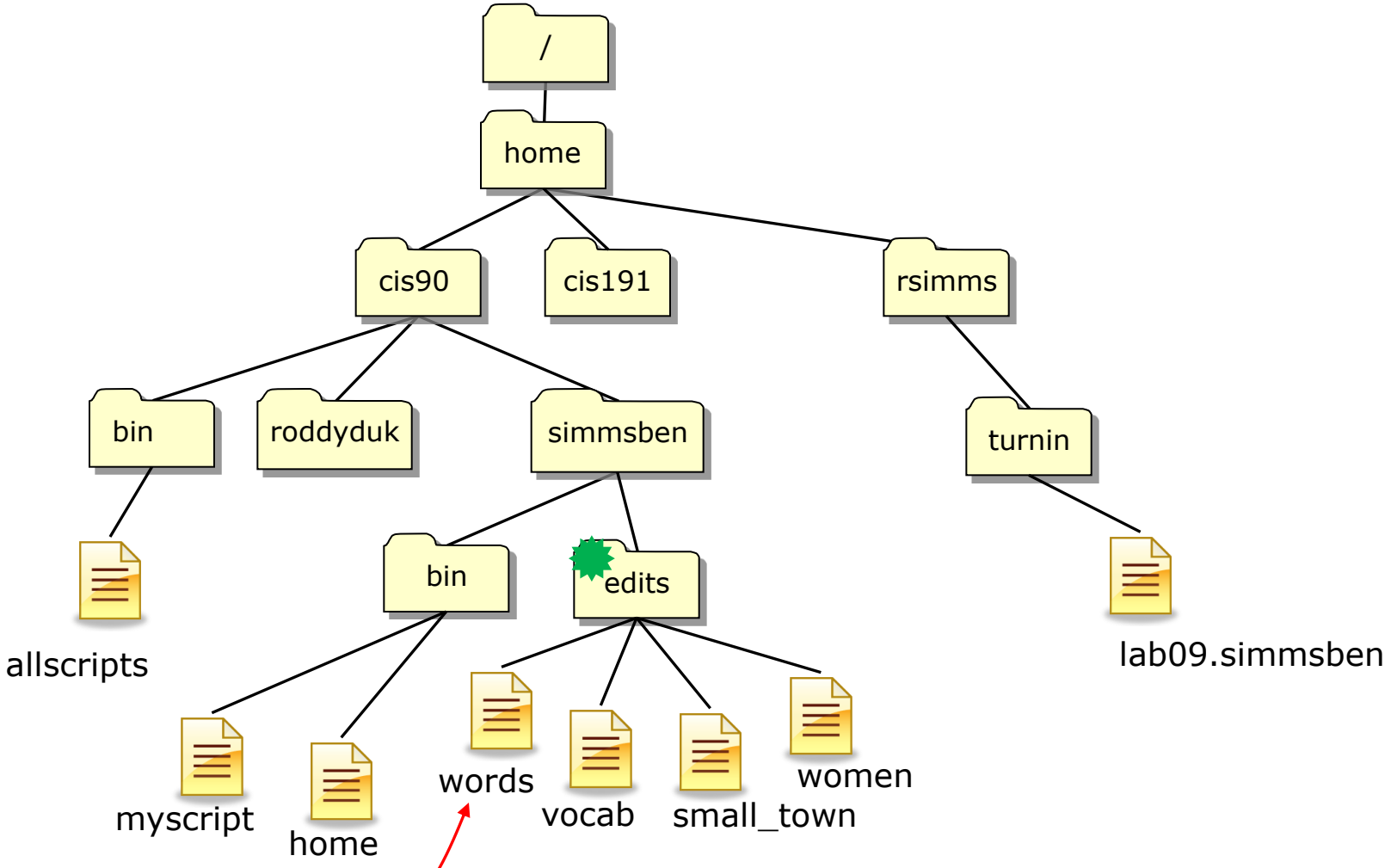
- Pathnames can be relative or absolute.



This is an absolute pathname

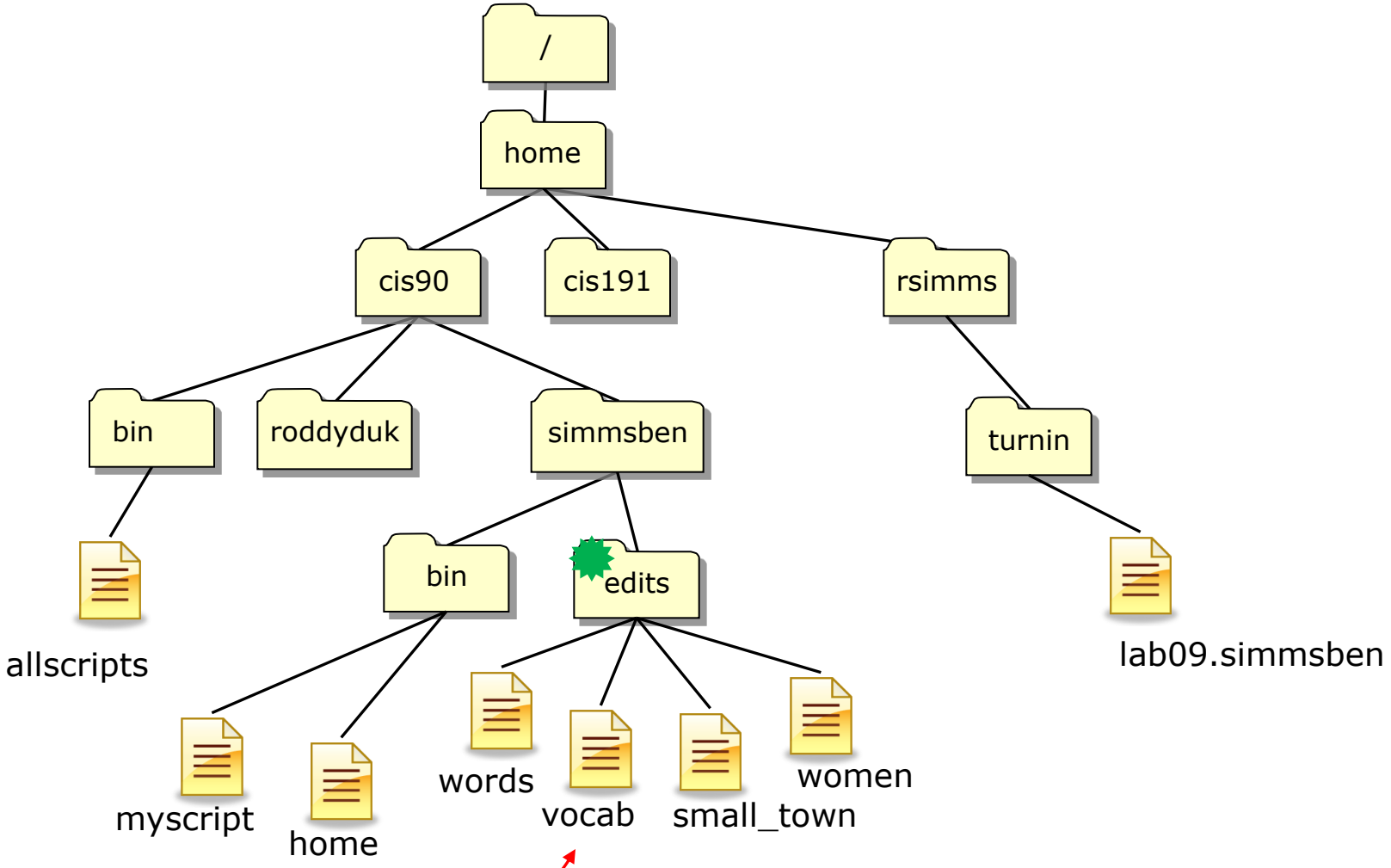
cat ../bin/home words vocab small_town woman > /home/rsimms/turnin/lab09.$LOGNAME

This is a relative pathname



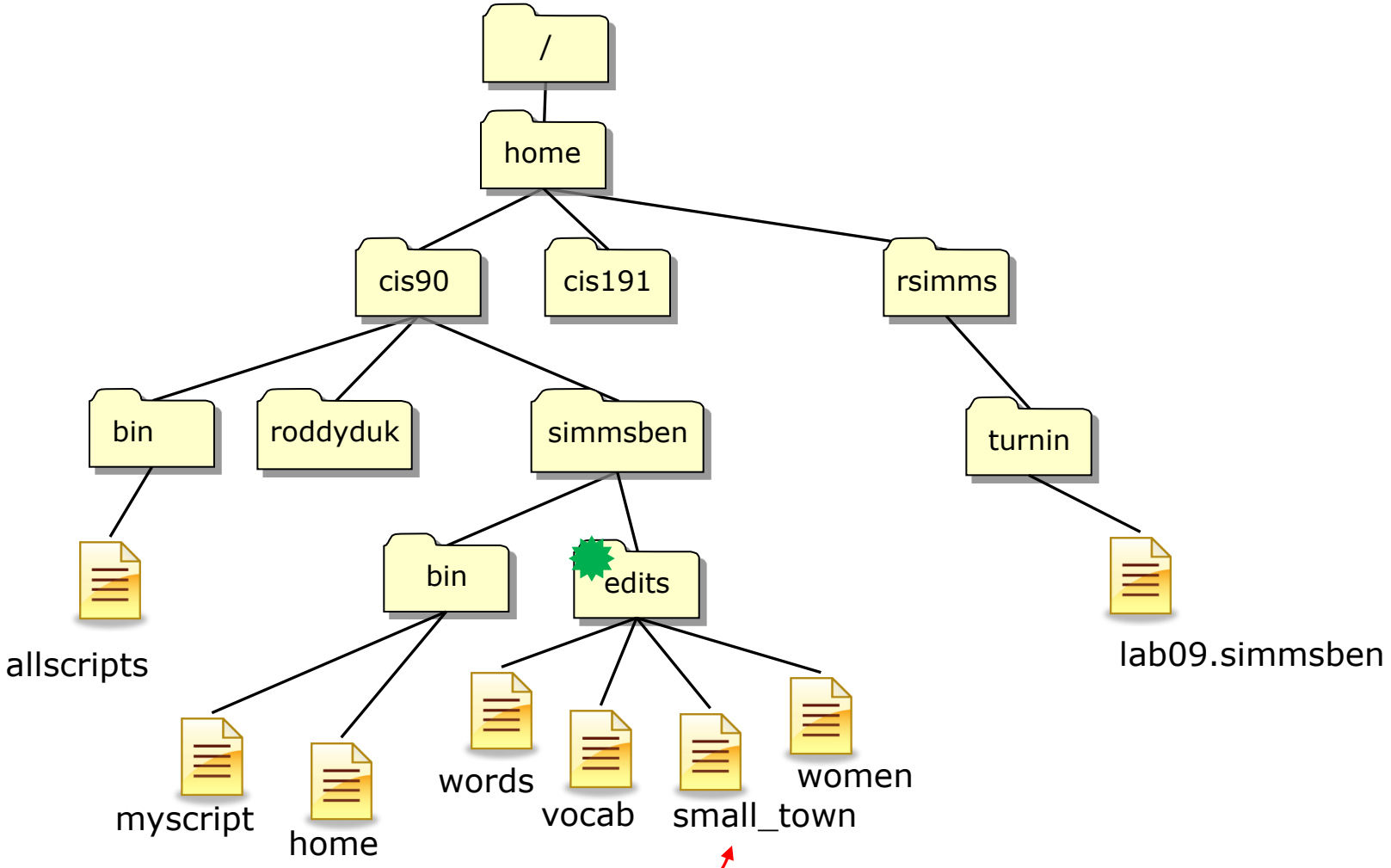
cat ../bin/home words vocab small_town woman > /home/rsimms/turnin/lab09.$LOGNAME

This is a relative pathname



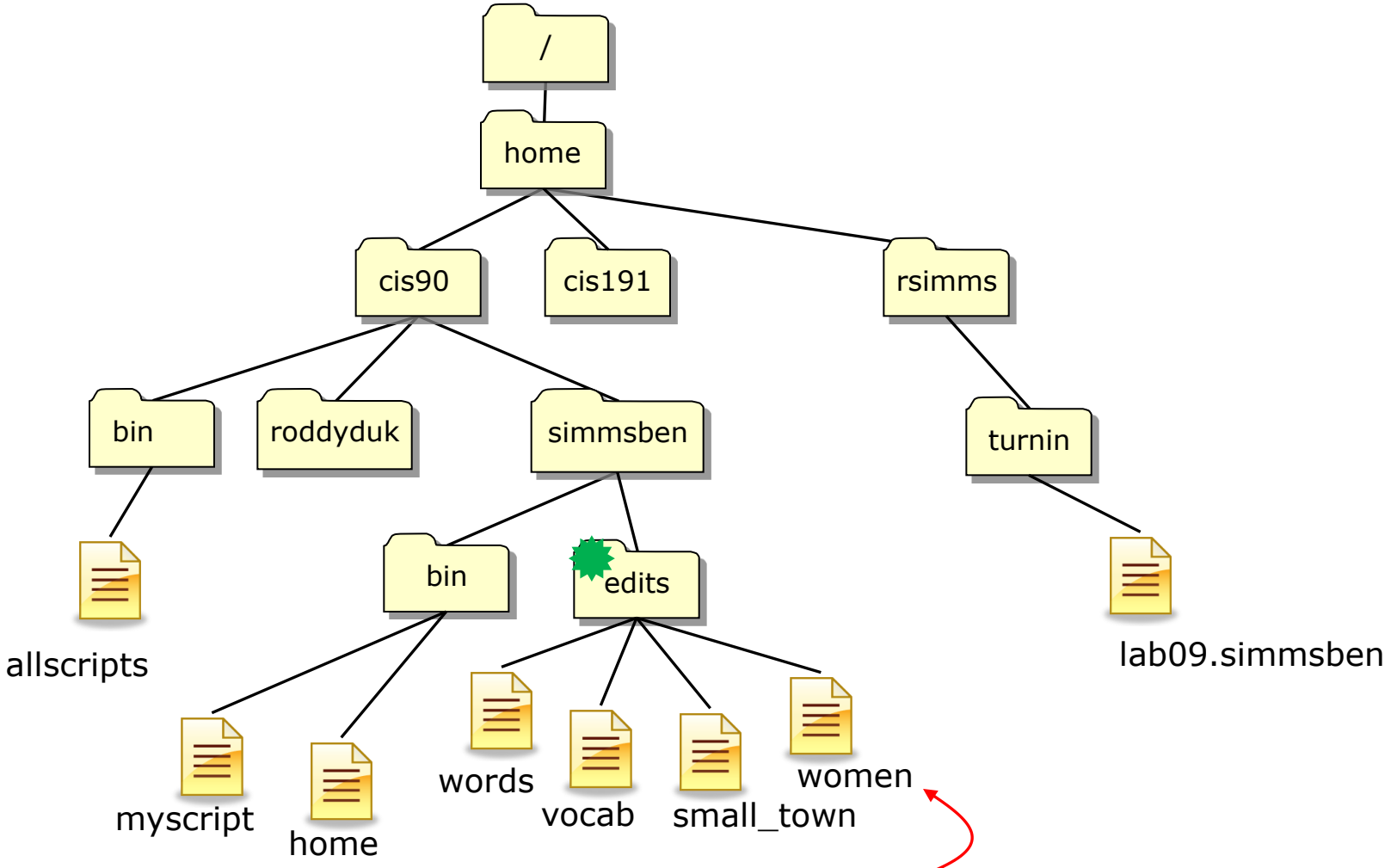
cat ../bin/home words vocab small_town woman > /home/rsimms/turnin/lab09.$LOGNAME

This is a relative pathname



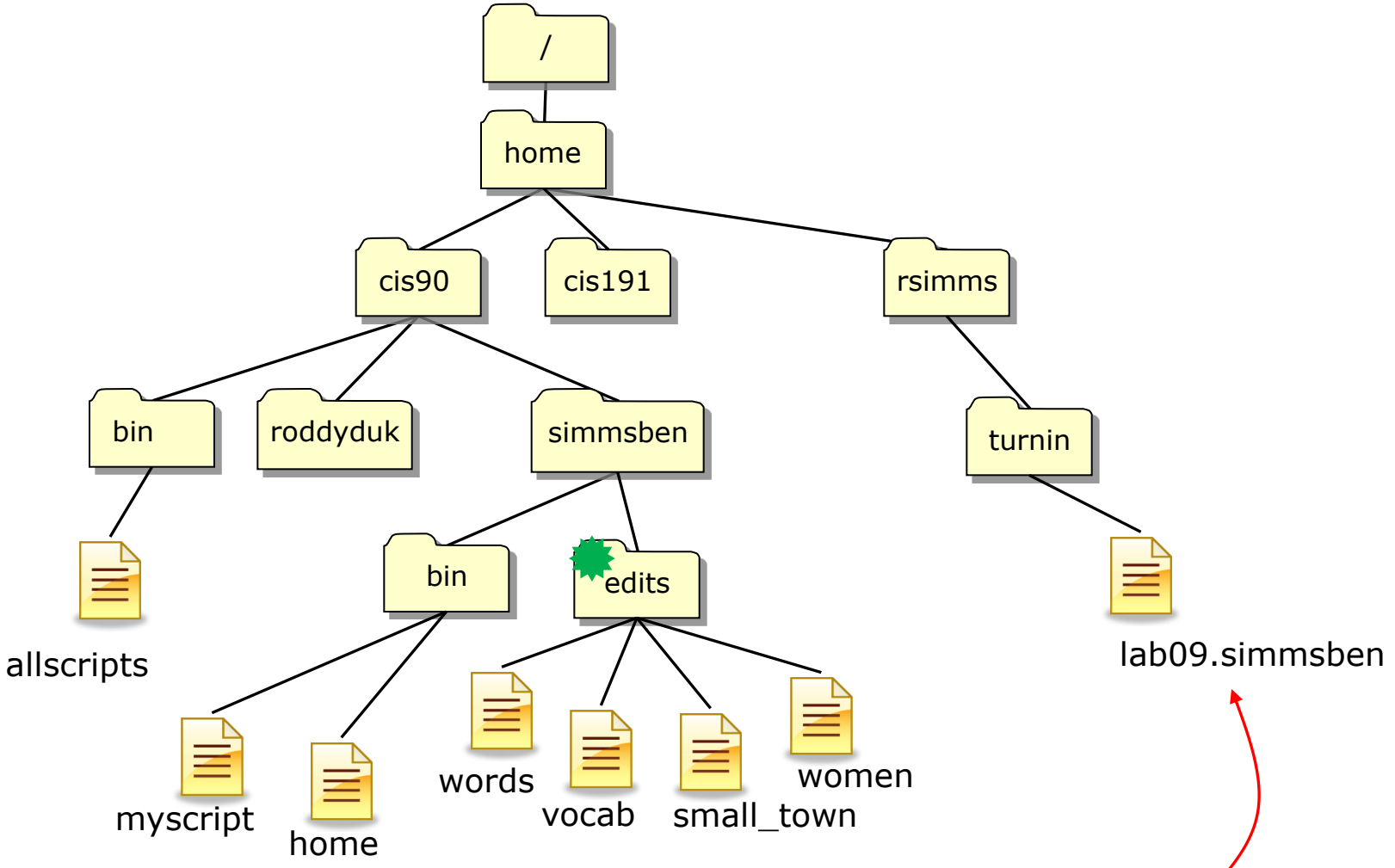
cat ../bin/home words vocab small_town woman > /home/rsimms/turnin/lab09.$LOGNAME

This is a relative pathname



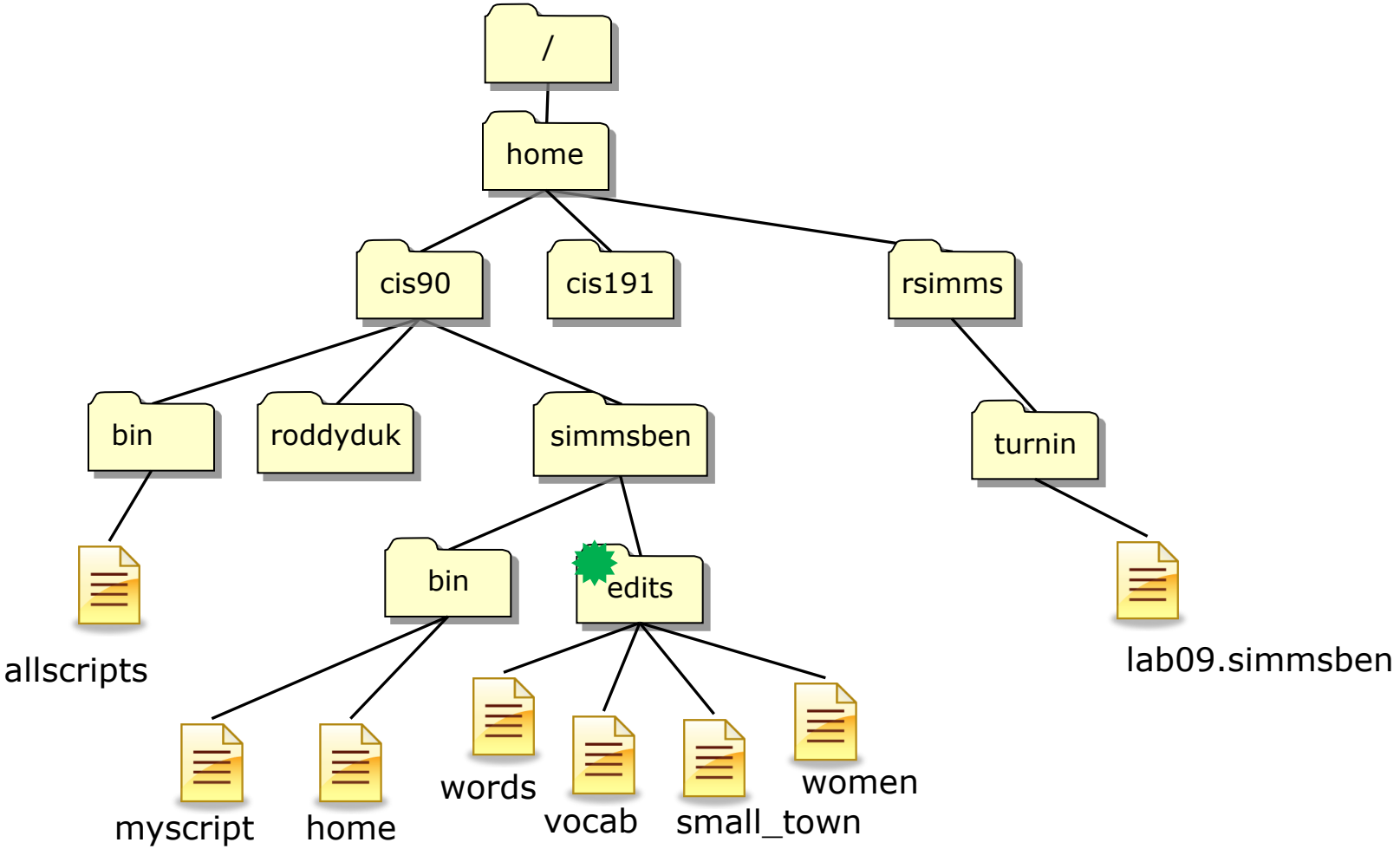
cat ../bin/home words vocab small_town woman > /home/rsimms/turnin/lab09.$LOGNAME

relative pathname



cat ../bin/home words vocab small_town woman > /home/rsimms/turnin/lab09.$LOGNAME

absolute pathname



Doing same thing in two steps

cat ../bin/home words vocab small_town woman > lab09
cp lab09 /home/rsimms/turnin/lab09.$LOGNAME

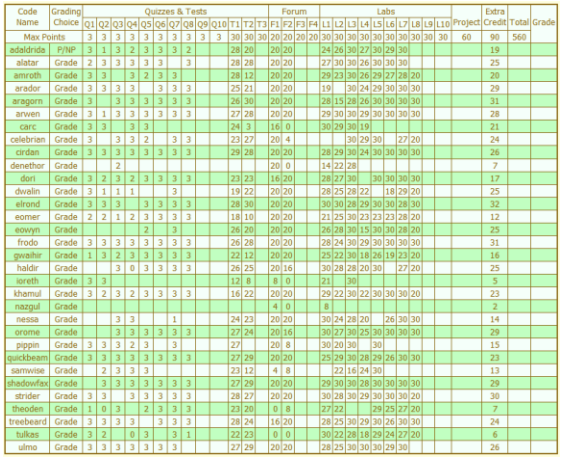
Housekeeping

Amgen Bike Race on Monday March 14th

http://www.tourofcalifornia-santacruz.com/race_day/transportation_and_parking
http://cabrillo.edu/services/marketing/amgen.html

Parking at or near Cabrillo on race day, Monday May 14th
Students and local residents are encouraged to take the shuttle, public transport or ride their bikes on race day. There will be a bike valet station at the Crocker Theater and shuttles will be running every 20 minutes from the following points and dropping folks on Soquel Ave near Cabrillo College road:

1. Capitola Mall 41st Ave. entrance
2. St. Joseph's Church, 435 Monterey Ave. Capitola
3. St. John's Church, 125 Canterbury Dr., Aptos
4. Coastlands Church, 280 State Park Dr., Aptos
5. Aptos Library, 7695 Soquel Dr. (This will be a "bridge" for any Santa Cruz County Metro passengers wishing to get dropped off at the Amgen drop off)
6. Amgen finish and festival drop off at the left hand turn lane in front of the tennis courts and shuttle turn around.



Code Name	Grading Choice	Quizzes & Tests													Forum				Labs										Project	Extra Credit	Total	Grade
		Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	T1	T2	T3	F1	F2	F3	F4	L1	L2	L3	L4	L5	L6	L7	L8	L9	L10				
Max Points		3	3	3	3	3	3	3	3	3	3	30	30	30	20	20	20	20	30	30	30	30	30	30	30	30	30	30	60	90	560	
adaldrida	P/NP	3	1	3	2	3	3	3	2			28	20		20	20			24	26	30	27	30	29	30					19			
alatar	Grade	2	3	3	3	3	3		3			28	28		20	20			27	30	30	26	30	30	30					25			
amroth	Grade	3	3		3	2	3	3				28	12		20	20			29	23	30	26	29	27	28	20				20			
arador	Grade	3	3	3	3		3	3	3			25	21		20	20			19		30	24	29	30	30	30				29			
aragorn	Grade	3		3	3	3	3	3	3			26	30		20	20			28	15	28	26	30	30	30	30				31			
arwen	Grade	3	1	3	3	3	3	3	3			27	28		20	20			29	30	30	29	30	30	30	30				28			
carc	Grade	3	3		3	3						24	3		16	0			30	29	30	19								21			
celebrian	Grade	3		3	3	2		3	3			23	27		20	4					30	29	30		27	20				24			
cirdan	Grade	3	3	3	3	3	3	3	3			29	28		20	20			28	29	30	24	30	30	30	30				26			
denethor	Grade			2											20	0			14	22	28									7			
dori	Grade	3	2	3	2	3	3	3	3			23	23		16	20			28	27	30		30	30	30	30				17			
dwalin	Grade	3	1	1	1			3				19	22		20	20			28	25	28	22		18	29	20				25			
elrond	Grade	3	3	3		3	3	3	3			28	30		20	20			30	30	28	29	30	30	28	30				32			
eomer	Grade	2	2	1	2	3	3	3	3			18	10		20	20			21	25	30	23	23	23	28	20				12			
eowyn	Grade					2		3				26	20		20	20			26	28	30	15	30	30	28	20				25			
frodo	Grade	3	3	3	3	3	3	3	3			26	28		20	20			28	24	30	29	30	30	30	30				31			
gwaihir	Grade	1	3	2	3	3	3	3	3			22	12		20	20			25	22	30	18	26	19	23	20				16			
haldir	Grade			3	0	3	3	3	3			26	25		20	16			30	28	28	20	30		27	20				25			
ioreth	Grade	3	3									12	8		8	0			21		30									5			
khamul	Grade	3	2	3	2	3	3	3	3			16	22		20	20			29	22	30	22	30	30	30	20				23			
nazgul	Grade														4	0			8											2			
nessa	Grade			3	3			1				24	23		20	20			30	24	28	20		26	30	30				14			
orome	Grade			3	3	3	3	3	3			27	24		20	16			30	27	30	25	30	30	30	30				29			
pippin	Grade	3	3	3	2	3		3				27			20	8			30	20	30		30							15			
quickbeam	Grade	3	3	3	3	3	3	3	3			27	29		20	20			25	29	30	28	29	26	30	30				23			
samwise	Grade		2	3	3	3						23	12		4	8			22	16	24	30								13			
shadowfax	Grade		3	3	3	3	3	3	3			27	29		20	20			29	30	30	28	30	30	30	30				29			
strider	Grade	3	3		3	3	3	3	3			28	27		20	20			30	28	30	29	30	30	30	20				30			
theoden	Grade	1	0	3		2	3	3	3			23	20		0	8			27	22			29	25	27	20				7			
treebeard	Grade	3	3	3	3		3	3	3			28	24		16	20			28	25	30	29	30	26	30	30				24			
tulkas	Grade	3	2		0	3		3	1			22	23		0	0			30	22	28	18	29	24	27	20				6			
ulmo	Grade	3	3	3	3	3	3	3	3			27	29		20	20			28	25	30	30	30	29	30					26			

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

Be sure to monitor you own progress using the Grades page of the course website

adaldrida: 88% (323 of 364 points)
alatar: 94% (344 of 364 points)
amroth: 90% (329 of 364 points)
arador: 90% (328 of 364 points)
aragorn: 100% (365 of 364 points)
arwen: 105% (383 of 364 points)
carc: 50% (184 of 364 points)
celebrian: 68% (251 of 364 points)
cirdan: 103% (378 of 364 points)
denethor: 25% (93 of 364 points)
dori: 89% (326 of 364 points)
dwalin: 78% (285 of 364 points)
elrond: 106% (386 of 364 points)
eomer: 80% (292 of 364 points)
eowyn: 88% (323 of 364 points)
frodo: 104% (380 of 364 points)

gwaihir: 80% (294 of 364 points)
haldir: 85% (310 of 364 points)
ioreth: 24% (90 of 364 points)
khamul: 92% (336 of 364 points)
nazgul: 3% (14 of 364 points)
nessa: 81% (296 of 364 points)
orome: 100% (366 of 364 points)
pippin: 54% (197 of 364 points)
quickbeam: 101% (370 of 364 points)
samwise: 44% (163 of 364 points)
shadowfax: 105% (383 of 364 points)
strider: 102% (373 of 364 points)
theoden: 61% (223 of 364 points)
treebeard: 99% (361 of 364 points)
tulkas: 71% (261 of 364 points)
ulmo: 94% (345 of 364 points)

12	5/2	**Quiz 9** **The Shell Environment** • Variables • Environment • .bash_profile • .bashrc **Materials** • Presentation slides (download) • bash startup files (download) **Assignment** • Lab 10 **CCC Confer** • Enter virtual classroom • Class archives	9 (Gillay)	Lab 9 5 posts
13	5/9	**Quiz 10** **Printing and Shell Scripting** • . and exec commands • UNIX printing **Materials** • Presentation slides (download) **Assignment** • Project **Extra Credit Lab** • Lab X1 (UNIX in Review) • Lab X2 (Pathnames) **CCC Confer** • Enter virtual classroom • Class archives	9 (Gillay)	Lab 10
14	5/16	**More Shell Scripting** • Examples • Command categories • scp command • tar command **Materials** • Presentation slides (download) **Assignment** • Project **CCC Confer** • Enter virtual classroom • Class archives	10 (Gillay)	
15	5/23	**File Transfer and Review** • Filezilla • gzip and gunzip • Project presentations • Project workshop **Materials** • Presentation slides (download) **Assignment** • Prep for Test #3 **CCC Confer** • Enter virtual classroom • Class archives	10 (Gillay)	Project
	5/30	**Test #3** **Time** • 1:00PM - 3:50PM **Materials** • Presentation slides (download) • Test (download) **CCC Confer** • Enter virtual classroom • Class archives		5 posts Lab X1 Lab X2

Endgame

If you like to be organized be sure and use the Calendar page on the website to plan your final weeks of class

Lab 10 due

Final project due

Final Exam (Test #3)
Five Posts due
Extra Credit Labs due

12	5/2	**Quiz 9** **The Shell Environment** • Variables • Environment • .bash_profile • .bashrc **Materials** • Presentation slides (download) • bash startup files (download) **Assignment** • Lab 10 **CCC Confer** • Enter virtual classroom • Class archives	9 (Gillay)	Lab 9 5 posts
13	5/9	**Quiz 10** **Printing and Shell Scripting** • . and exec commands • UNIX printing **Materials** • Presentation slides (download) **Assignment** • Project **Extra Credit Lab** • Lab X1 (UNIX in Review) • Lab X2 (Pathnames) **CCC Confer** • Enter virtual classroom • Class archives	9 (Gillay)	Lab 10
14	5/16	**More Shell Scripting** • Examples • Command categories • scp command • tar command **Materials** • Presentation slides (download) **Assignment** • Project **CCC Confer** • Enter virtual classroom • Class archives	10 (Gillay)	
15	5/23	**File Transfer and Review** • Filezilla • gzip and gunzip • Project presentations • Project workshop **Materials** • Presentation slides (download) **Assignment** • Prep for Test #3 **CCC Confer** • Enter virtual classroom • Class archives	10 (Gillay)	Project
	5/30	**Test #3** **Time** • 1:00PM - 3:50PM **Materials** • Presentation slides (download) • Test (download) **CCC Confer** • Enter virtual classroom • Class archives		5 posts Lab X1 Lab X2

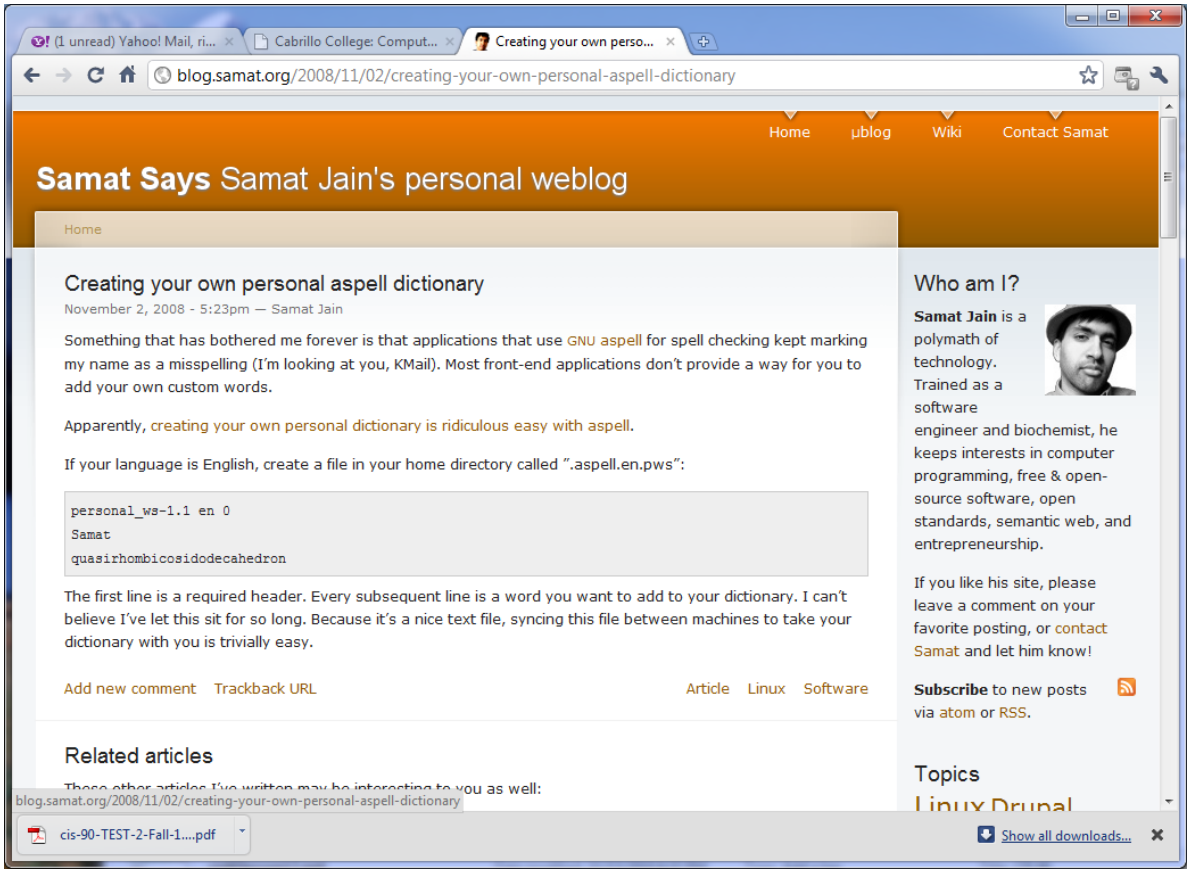
Final Exam

Can **not** be taken online using CCC Confer

It will be held in room 2501 on May 30th from 1:00 to 3:50PM

Ayshire moshpit and personal dictionaries

aspell command



Googling "linux aspell personal dictionary" yields this page

Bingo! Thank you Samat Jain

spell command

```
/home/cis90ol/simmsben $ echo Benji lives in Soquel > address
/home/cis90ol/simmsben $ cat address
Benji lives in Soquel
/home/cis90ol/simmsben $ spell address
Soquel

/home/cis90ol/simmsben $ echo "personal_ws-1.1 en 0" > .aspell.en.pws
/home/cis90ol/simmsben $ echo Soquel >> .aspell.en.pws
/home/cis90ol/simmsben $ spell address
/home/cis90ol/simmsben $
```

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

This is FYI and not required for Lab 9

Make a Personal Dictionary

```
cd
echo "personal_ws-1.1 en 0" >  .aspell.en.pws
echo "moshpit" >> .aspell.en.pws
echo "Ayshire" >> .aspell.en.pws
cat .aspell.en.pws

cd edits/
spell small_town
```

Note: You should still leave the two words Ayshire and mashpit (or moshpit) in the file words when you submit Lab 9

vi

Best Practice - /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

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.

/bin/mail and vi

Try it!

Use /bin/mail and send me (rsimms) a message that you have made or corrected using the ~vi command

cc: yourself so you can verify what you sent.

final project

Final Project

Cabrillo College

CIS 90 Final Project

Developing a bash script
Spring 2012

Final Project

For the final project you will be writing custom front-ends to your favorite Linux commands. To do this you will write a shell script that interacts with the user to get input, then use that input to call a Linux command. You will start with a template that you can modify and extend.

Forum

Use the forum to brainstorm script ideas, clarify requirements, and get help if you are stuck. When you have tested your script and think it is bug free then use the forum to ask others to test it some more. Post any valuable tips or lessons learned as well. Forum is at: http://opus.cabrillo.edu/forum/viewforum.php?f=46

Commands

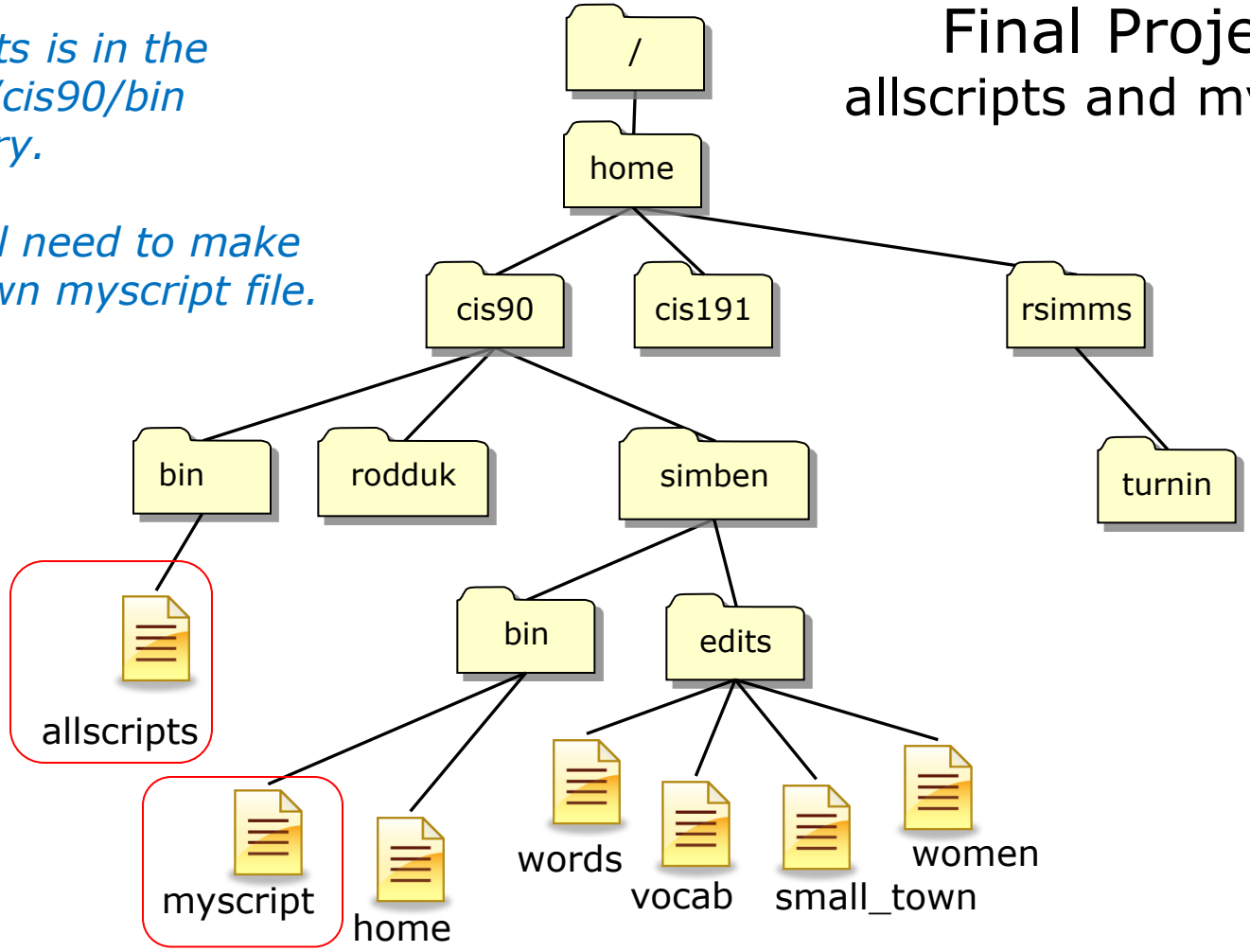
.	echo	lpstat	sort
at	env	ls	spell
banner	exit	mail	su
bash	export	man	tail
bc	file	mesg	tee
cal	find	mkdir	touch
cancel	finger	more	type
cat	grep	mv	umask
cd	head	passwd	uname
chgrp	history	ps	unset
chmod	id	pwd	vi
chown	jobs	rm	wc
clear	kill	rmdir	who
cp	ln	set	write
date	lp/lpr	sleep	xwd

You now have the necessary skills to begin the final project!

Final Project
allscripts and myscript

allscripts is in the /home/cis90/bin directory.

You will need to make your own myscript file.




```
/home/cis90ol/roddyduk $ ls -l /home/cis90/bin/allscripts  bin/myscript
-rwxr-xr-x 1 simben90 cis90 4296 Nov 13 13:07 bin/myscript
-rwxr-xr-x 1 rsimms   staff 4381 Nov 13 18:17 /home/cis90/bin/allscripts
```

Final Project
/home/cis90/bin/allscripts

```
#!/bin/bash
#
# menu: A simple menu template
#
while true
do
        clear
        echo -n "
        ***************************************************************
        *                Spring 2012 CIS 90 Online Projects           *
        ***************************************************************
        1) Ana          10) Dieskau     19) Mason       28) Raven
```

More menu lines (reduced in size to fit on page)

Print a menu with each student's name

```
     Enter Your Choice: "
        read RESPONSE
        case $RESPONSE in
          1)    # Ana
                        /home/cis90/varana/bin/myscript
                        ;;
```

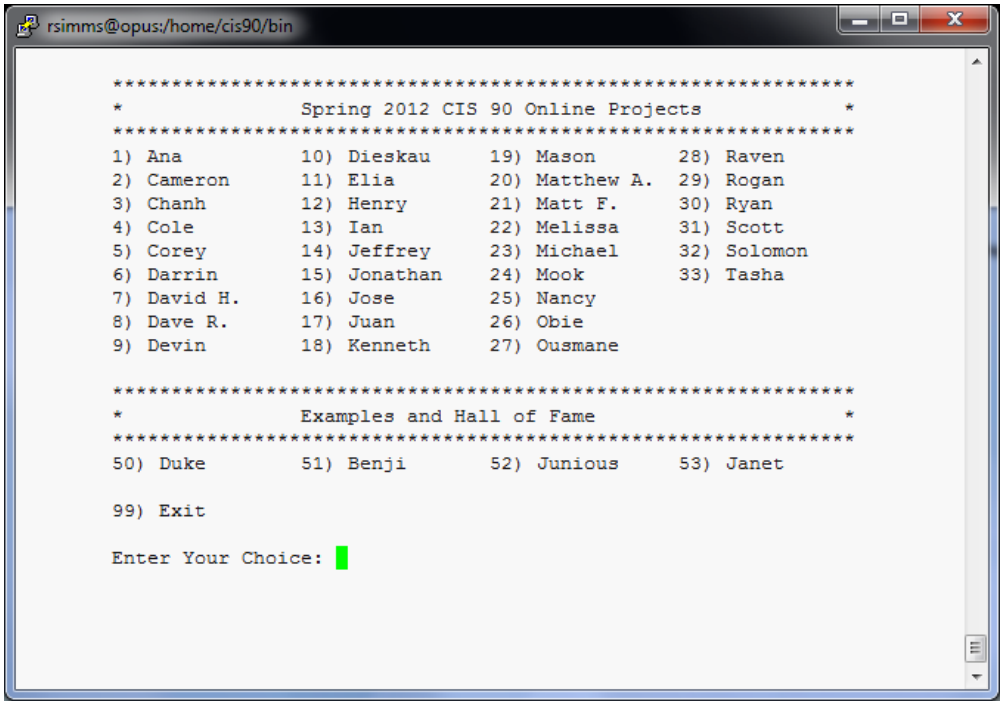
The rest of the pathnames to student's myscript files (reduced in size to fit on page)

For every student name in the menu, there is a pathname to that student's myscript file

```
          99)   exit 0
                        ;;
          *)    echo "Please enter a number from above"
                        ;;
        esac
        echo -n "Hit the Enter key to return to menu "
        read dummy
done
```

Final Project
allscripts (continued)

Running **/home/cis90/bin/allscripts** looks like this



*This script has been updated with everyone's name and pathnames to each student's **myscript** file*

Final Project
myscript

/home/cis90/$LOGNAME/bin/myscript

simmsben@opus:~/bin

```
#!/bin/bash
#
# menu: A simple menu template
#
while true
do
clear
echo -n "
CIS 90 Final Project
1) Task 1
2) Task 2
3) Task 3
4) Task 4
5) Task 5
6) Exit
Enter Your Choice: "
read RESPONSE
case $RESPONSE in
1) # Commands for Task 1
;;
2) # Commands for Task 2
;;
3) # Commands for Task 3
;;
4) # Commands for Task 4
;;
5) # Commands for Task 5
;;
6) exit 0
;;
*) echo "Please enter a number between 1 and 6"
;;
esac
echo -n "Hit the Enter key to return to menu "
read dummy
done
                                                    36,1          All
```

*Your initial **myscript** file will look like this in vi*

vi understands shell scripts and will use color syntax styling.

*Every student needs to create a **myscript** file in their bin directory.*

*Use vi to create the **myscript** file and copy and paste the template code from the Final Project into it.*

Final Project

/home/cis90/$LOGNAME/bin/myscript

Getting Started

1) On Opus, cd to your bin directory and enter:
   vi myscript
   then type **i** to enter insert mode

2) In your web browser, view the CIS 90 calendar page and click on the project link for Lesson 15. Select the template code and copy it to the clipboard.

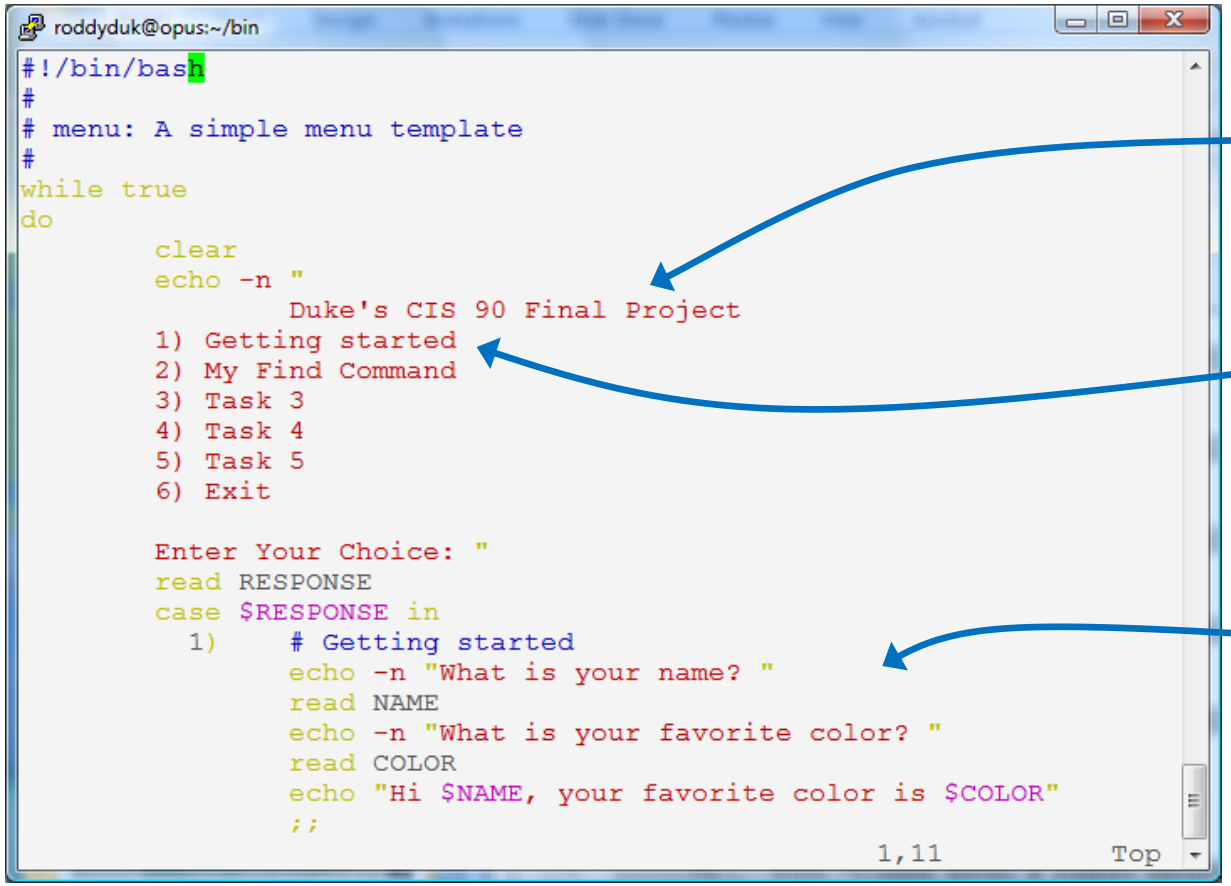
3) Click back on the vi session and click the right mouse button to paste the template code.

4) Save the code with **Esc** and the **:wq**

5) Give myscript execute permissions with **chmod +x myscript**

Final Project

/home/cis90/$LOGNAME/bin/myscript



Customize your menu title

Add a menu entry

Add some sample dialog code using variables

Final Project

/home/cis90/$LOGNAME/bin/myscript

A new command

```
read RESPONSE
case $RESPONSE in
  1)    # Getting started
        echo -n "What is your name? "
        read NAME
        echo -n "What is your favorite color? "
        read COLOR
        echo "Hi $NAME, your favorite color is $COLOR"
        ;;
```

another new command

Final Project

/home/cis90/$LOGNAME/bin/myscript

case statement begins here

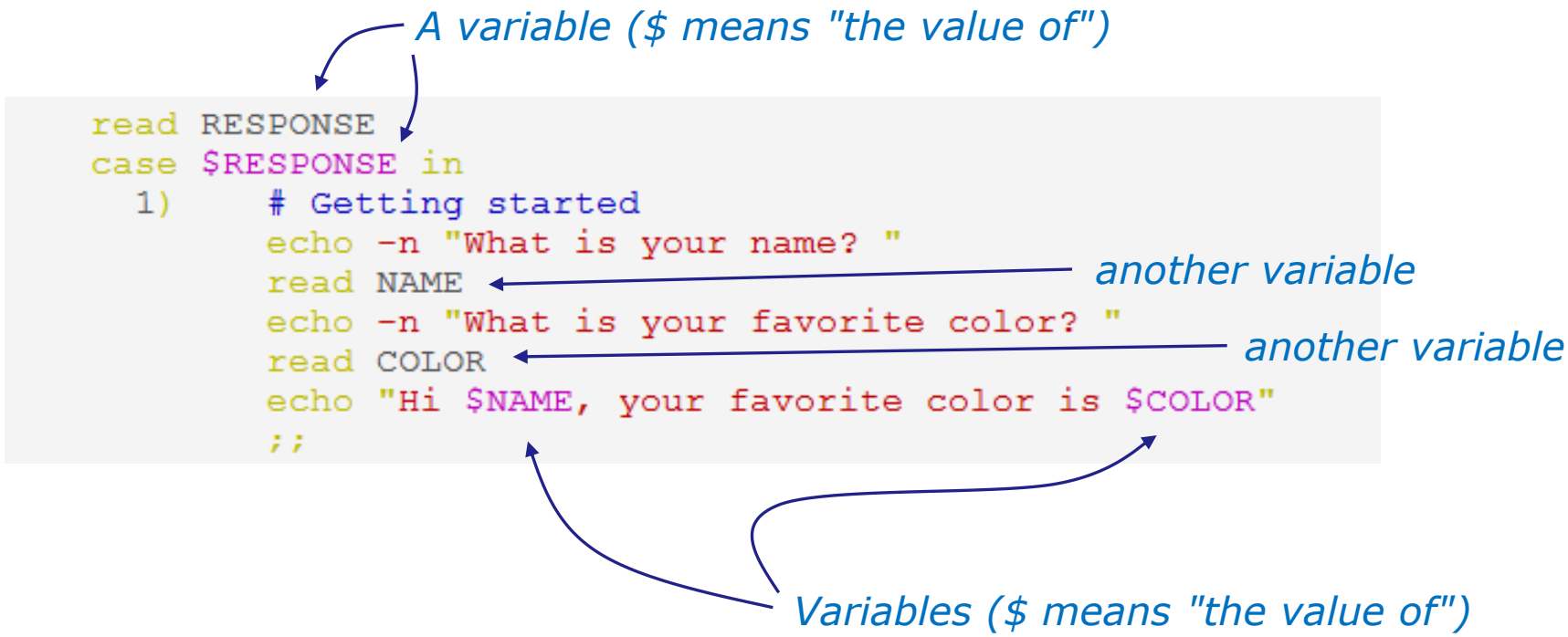
```
read RESPONSE
case $RESPONSE in
  1)    # Getting started
        echo -n "What is your name? "
        read NAME
        echo -n "What is your favorite color? "
        read COLOR
        echo "Hi $NAME, your favorite color is $COLOR"
        ;;
```

First case ends here

First case of case statement starts here

Final Project

/home/cis90/$LOGNAME/bin/myscript



Final Project

/home/cis90/$LOGNAME/bin/myscript

```
read RESPONSE
case $RESPONSE in
  1)    # Getting started
        echo -n "What is your name? "
        read NAME
        echo -n "What is your favorite color? "
        read COLOR
        echo "Hi $NAME, your favorite color is $COLOR"
        ;;
```

Comments begin with a #

Final Project

/home/cis90/$LOGNAME/bin/myscript

roddyduk@opus:~/bin

```
#!/bin/bash
#
# menu: A simple menu template
#
while true
do
        clear
        echo -n "
                Duke's CIS 90 Final Project
        1) Getting started
        2) My Find Command
        3) Task 3
        4) Task 4
        5) Task 5
        6) Exit

        Enter Your Choice: "
        read RESPONSE
        case $RESPONSE in
          1)    # Getting started
                echo -n "What is your name? "
                read NAME
                echo -n "What is your favorite color? "
                read COLOR
                echo "Hi $NAME, your favorite color is $COLOR"
                ;;
                                                1,11          Top
```

Customize your menu title

Customize the first menu entry

Add this sample dialog code using variables

*When finished, test both the **myscript** and **allscripts** "commands"*

Shell Variables

Shell Variables

- Shell variables names consist of alpha-numeric characters.
- Variables defined by the Operating System are uppercase, e.g. TERM, PS1, PATH
- The **set** command will display the shell's current variables and their values.
- Shell variables are initialized using the assignment operator:
  For example: **TERM=vt100**
  Note: Quotes must be used for white space: **VALUE="any value"**
- Variables may be viewed using the echo command: **echo $TERM**
  The $ in front of a variable name denotes the value of that variable.
- To remove a variable, use the unset command: **unset PS1**
- Shell variables hold their values for the duration of the session i.e. until the shell is exited

Shell Variables

Showing values of variables

Use: **echo $***varname*
to show value of a variable

```
[rsimms@nosmo ~]$ echo $PATH
/usr/kerberos/bin:/usr/local/bin:/bin:/usr/bin:/usr/X11R6/bin:/home/rsimms/bin

[rsimms@nosmo ~]$ echo $TERM
xterm

[rsimms@nosmo ~]$ echo $HOME
/home/rsimms

[rsimms@nosmo ~]$ echo $PS1
[\u@\h \W]\$
```

Using the echo command to show the values of several environment variables

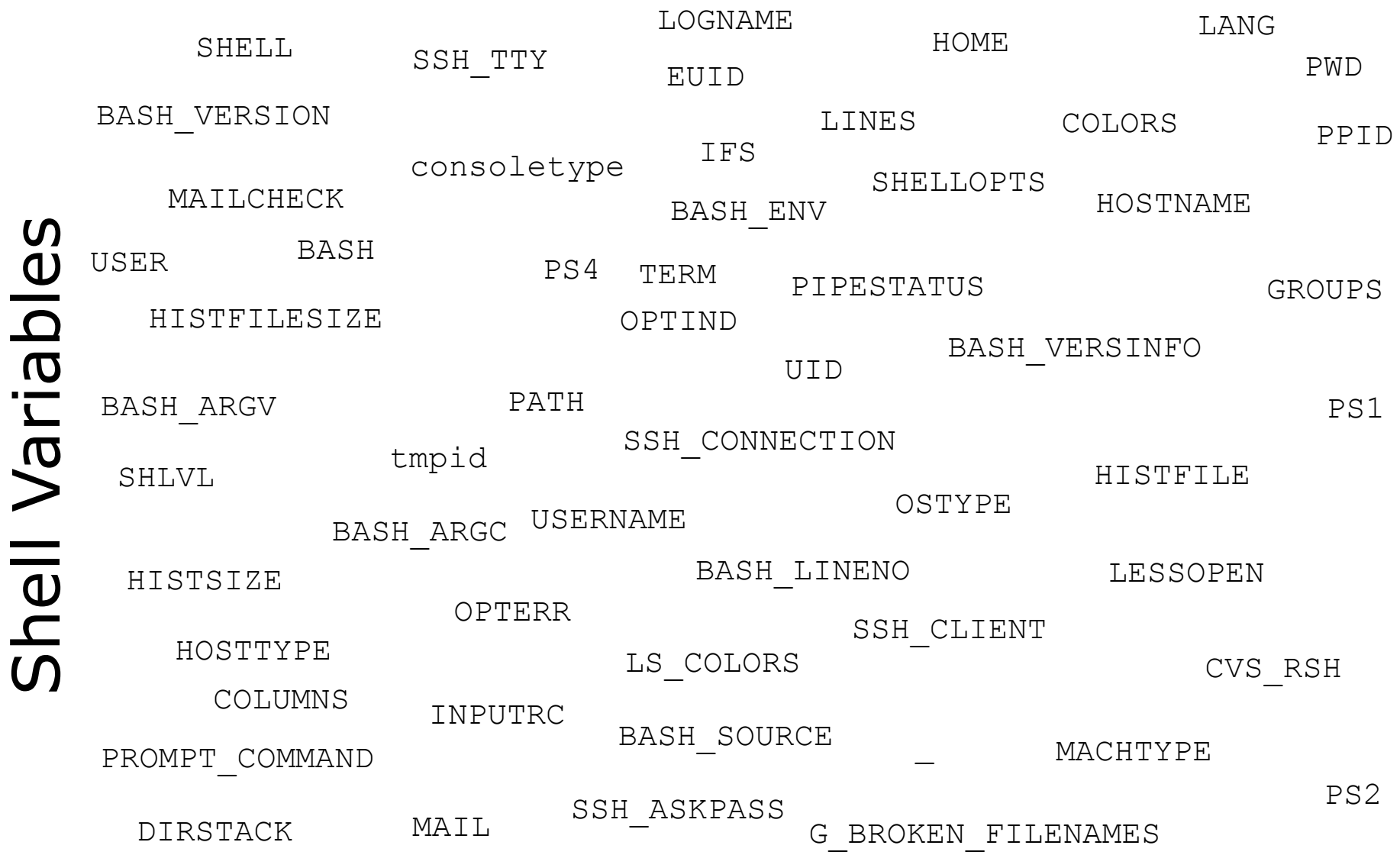
Shell Variables

Setting the values of variables

Use: *varname*=*value*
(no spaces please around the =)

```
[rsimms@nosmo ~]$ PS1="By your command >"
By your command >
By your command >PS1="What can I do for you $LOGNAME? "
What can I do for you rsimms?
What can I do for you rsimms?


/home/cis90ol/simmsben/bin $ river="The Amazon"
/home/cis90ol/simmsben/bin $ echo $river
The Amazon
/home/cis90ol/simmsben/bin $ echo river
river
```

Shell Variables



*See all shell variables by typing **set***

Shell Variables

```
/home/cis90/simmsben/Poems $set
BASH=/bin/bash
BASH_ARGC=()
BASH_ARGV=()
BASH_ENV=/home/cis90/simmsben/.bashrc
BASH_LINENO=()
BASH_SOURCE=()
BASH_VERSINFO=([0]="3" [1]="2" [2]="25" [3]="1"
[4]="release" [5]="i686-redhat-linux-gnu")
BASH_VERSION='3.2.25(1)-release'
COLORS=/etc/DIR_COLORS.xterm
COLUMNS=80
CVS_RSH=ssh
DIRSTACK=()
EUID=1160
GROUPS=()
G_BROKEN_FILENAMES=1
HISTFILE=/home/cis90/simmsben/.bash_history
HISTFILESIZE=1000
HISTSIZE=1000
HOME=/home/cis90/simmsben
HOSTNAME=opus.cabrillo.edu
HOSTTYPE=i686
IFS=$' \t\n'
IGNOREEOF=10
INPUTRC=/etc/inputrc
LANG=en_US.UTF-8
LESSOPEN='|/usr/bin/lesspipe.sh %s'
LINES=24
LOGNAME=simmsben
```

The set command, with no arguments, will show all shell variables

```
LS_COLORS='no=00:fi=00:di=00;34:ln=00;36:pi=40;33:so=00;35
:bd=40;33;01:cd=40;33;01:or=01;05;37;41:mi=01;05;37;41:ex=
00;32:*.cmd=00;32:*.exe=00;32:*.com=00;32:*.btm=00;32:*.ba
t=00;32:*.sh=00;32:*.csh=00;32:*.tar=00;31:*.tgz=00;31:*.a
rj=00;31:*.taz=00;31:*.lzh=00;31:*.zip=00;31:*.z=00;31:*.Z
=00;31:*.gz=00;31:*.bz2=00;31:*.bz=00;31:*.tz=00;31:*.rpm=
00;31:*.cpio=00;31:*.jpg=00;35:*.gif=00;35:*.bmp=00;35:*.x
bm=00;35:*.xpm=00;35:*.png=00;35:*.tif=00;35:'
MACHTYPE=i686-redhat-linux-gnu
MAIL=/var/spool/mail/simmsben
MAILCHECK=60
OLDPWD=/home/cis90/simmsben
OPTERR=1
OPTIND=1
OSTYPE=linux-gnu
PATH=/usr/kerberos/bin:/usr/local/bin:/bin:/usr/bin:/home/
cis90/simmsben/../bin:/home/cis90/simmsben/bin:.
PIPESTATUS=([0]="0")
PPID=26514
PROMPT_COMMAND='echo -ne
"\033]0;${USER}@${HOSTNAME%%.*}:${PWD/#$HOME/~}"; echo -ne
"\007"'
PS1='$PWD $'
PS2='> '
PS4='+ '
PWD=/home/cis90/simmsben/Poems
SHELL=/bin/bash
SHELLOPTS=braceexpand:emacs:hashall:histexpand:ignoreeof:i
nteractive-comments:monitor
SHLVL=1
SSH_ASKPASS=/usr/libexec/openssh/gnome-ssh-askpass
TERM=xterm
UID=1160
USER=simmsben
USERNAME=
_=env
consoletype=pty
```

Shell Variables

1

```
/home/cis90ol/simmsben/bin $ echo $defrost $ac $fan

/home/cis90ol/simmsben/bin $
```

the value of a variable that has not been created is null

2

```
/home/cis90/roddyduk $ defrost=on
/home/cis90/roddyduk $ ac=off
/home/cis90/roddyduk $ fan=medium
```

create some new shell variables and assign values

3

```
/home/cis90/roddyduk $ echo $defrost $ac $fan
on off medium

/home/cis90/roddyduk $ echo defrost ac fan
defrost ac fan
```

*print the **values** of the shell variables*

*print the **names** of the shell variables*

Class Exercise

Create and intialize three new variables:

defrost=on
ac=off
fan=medium

Show the names of the variables:

echo defrost ac fan

Show the values of the variables:

echo $defrost $ac $fan

Shell Variables

```
/home/cis90/roddyduk $ unset defrost
/home/cis90/roddyduk $ unset ac fan
```

delete the new shell variables

```
/home/cis90/roddyduk $ echo $defrost $ac $fan

/home/cis90/roddyduk $
```

Non-existing variables have null values

Class Exercise

Delete your three new variables:

unset defrost
unset ac fan

Show the names of the variables:

echo defrost ac fan

Show the values of the variables:

echo $defrost $ac $fan

Shell Variables

```
/home/cis90/roddyduk $ defrost=on
/home/cis90/roddyduk $ ac=off
/home/cis90/roddyduk $ fan=medium
/home/cis90/roddyduk $ set
```

Note: Any new variables you initialize will show up in the output of the set command

font reduced for the other variables to fit on slide

Shell Variables

Using grep to find a variable in the output of the set command

```
/home/cis90/roddyduk $ set | grep defrost
defrost=on
```

In this example:

- *the **set** command outputs all the shell variables to stdout*
- *which is piped to the stdin of the grep command*
- *the **grep** command examines each line one at a time and only outputs the lines containing the string "defrost"*

Shell Variables

Variables are often used in scripts when you need a placeholder to store some data

1

```
/home/cis90/roddyduk $ vi funscript
/home/cis90/roddyduk $ cat funscript
#!/bin/bash
echo -n "Turn the Air Conditioning on or off? "
read ac
echo "Air Conditioning set to $ac"
exit
```

Create a script that uses a variable named "ac" to hold the status of an air conditioner. Prompt the user and input what they type into the this variable.

2

```
/home/cis90/roddyduk $ chmod +x funscript
```

Add execute permissions so the script can be run

3

```
/home/cis90/roddyduk $ ./funscript
Turn the Air Conditioning on or off? off
Air Conditioning set to off
```

Run the script

Class Exercise

Now make this little user dialog script:

vi funscript

insert the following lines then save

```
#!/bin/bash
echo -n "Turn the Air Conditioning on or off? "
read ac
echo "Air Conditioning set to $ac"
exit
```

chmod +x funscript

./funscript

Environment Variables

Environment Variables

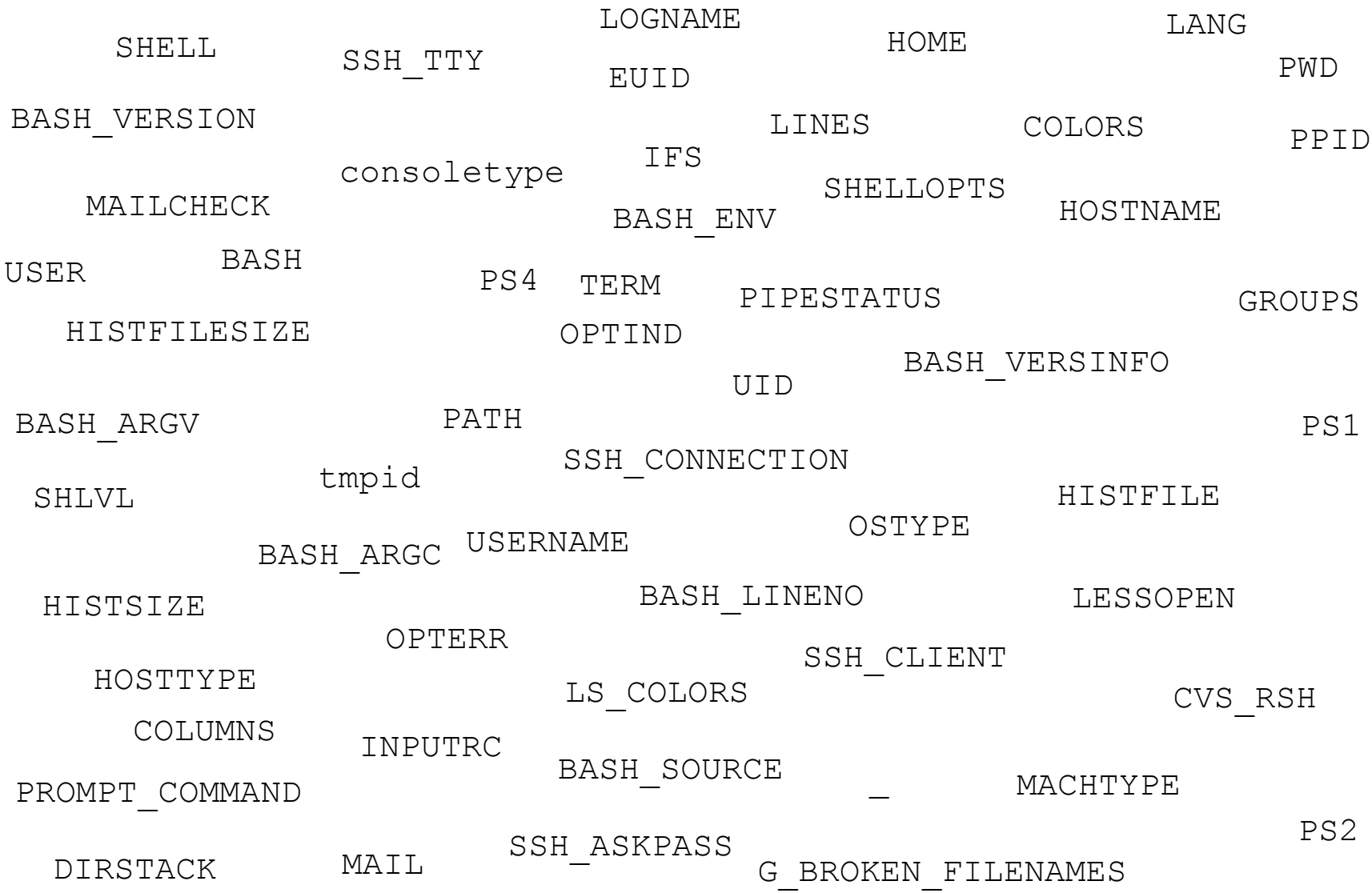
- Environment variables are a special subset of the shell variables.
- Environment variables are shell variables that have been *exported*.
- The **env** command will display the current environment variables and their values. Using the **export** command with no arguments will also show all the environment variables.
- The **export** command is used to make a shell variable into an environment variable.

  dog=benji; export dog
  or **export dog=benji**

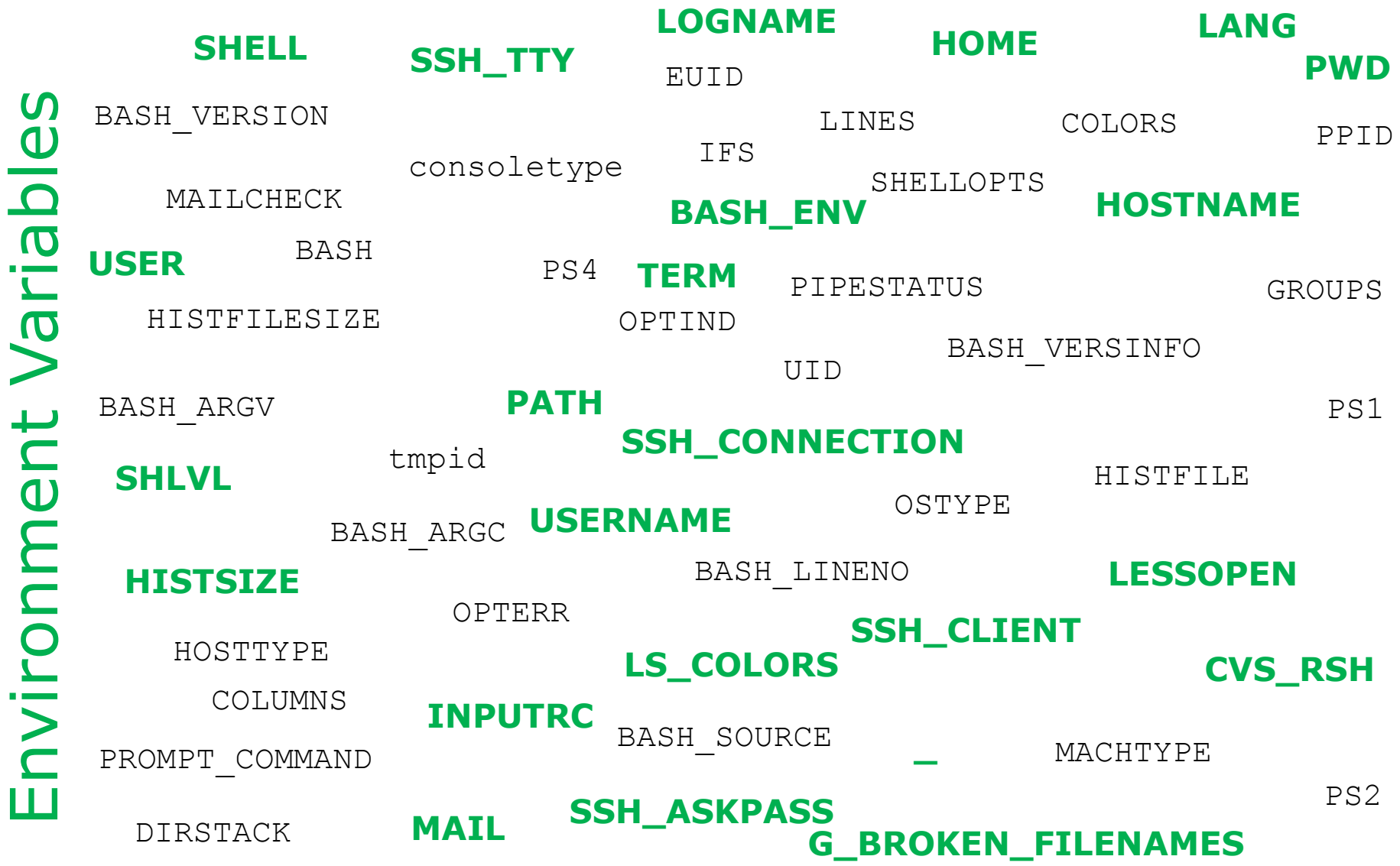
- The **export -n** command is used to make an environment variable back into a normal shell variable. E.g. **export -n dog** makes dog back into a regular shell variable.

Child processes are provided copies of the parent's environment variables. Any changes made by the child will not effect the parent's copies.

Shell Variables



*Use the **set** command to show all shell variables*



*Use the **env** to see which of the shell variables have been exported and therefore environment variables (shown in bold/green above)*

Shell (Environment) Variables

Some famous environment variables

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

Shell (Environment) Variables

env command – show all environment variables

```
[roddyduk@opus ~]$ env
HOSTNAME=opus.cabrillo.edu
SHELL=/bin/bash
TERM=xterm
HISTSIZE=1000
SSH_CLIENT=63.249.103.107 20807 22
SSH_TTY=/dev/pts/0
USER=roddyduk
LS_COLORS=no=00:fi=00:di=00;34:ln=00;36:pi=40;33:so=00;35:bd=40;33;01:cd=40;33;01:or=01;05;37;41:mi=01;05
;37;41:ex=00;32:*.cmd=00;32:*.exe=00;32:*.com=00;32:*.btm=00;32:*.bat=00;32:*.sh=00;32:*.csh=00;32:*.tar=
00;31:*.tgz=00;31:*.arj=00;31:*.taz=00;31:*.lzh=00;31:*.zip=00;31:*.z=00;31:*.Z=00;31:*.gz=00;31:*.bz2=00
;31:*.bz=00;31:*.tz=00;31:*.rpm=00;31:*.cpio=00;31:*.jpg=00;35:*.gif=00;35:*.bmp=00;35:*.xbm=00;35:*.xpm=
00;35:*.png=00;35:*.tif=00;35:
USERNAME=
PATH=/usr/kerberos/bin:/usr/local/bin:/bin:/usr/bin:/home/cis90/roddyduk/../bin:/home/cis90/roddyduk/bin:
.
MAIL=/var/spool/mail/roddyduk
PWD=/home/cis90/roddyduk
INPUTRC=/etc/inputrc
LANG=en_US.UTF-8
fan=medium
SSH_ASKPASS=/usr/libexec/openssh/gnome-ssh-askpass
HOME=/home/cis90/roddyduk
SHLVL=2
BASH_ENV=/home/cis90/roddyduk/.bashrc
LOGNAME=roddyduk
CVS_RSH=ssh
SSH_CONNECTION=63.249.103.107 20807 207.62.186.9 22
LESSOPEN=|/usr/bin/lesspipe.sh %s
G_BROKEN_FILENAMES=1
_=/bin/env
```

The env command by itself will list all the environment (exported) variables

Shell (Environment) Variables

export command – show all exported variables

```
[roddyduk@opus ~]$ export
declare -x BASH_ENV="/home/cis90/roddyduk/.bashrc"
declare -x CVS_RSH="ssh"
declare -x G_BROKEN_FILENAMES="1"
declare -x HISTSIZE="1000"
declare -x HOME="/home/cis90/roddyduk"
declare -x HOSTNAME="opus.cabrillo.edu"
declare -x INPUTRC="/etc/inputrc"
declare -x LANG="en_US.UTF-8"
declare -x LESSOPEN="|/usr/bin/lesspipe.sh %s"
declare -x LOGNAME="roddyduk"
declare -x
LS_COLORS="no=00:fi=00:di=00;34:ln=00;36:pi=40;33:so=00;35:bd=40;33;01:cd=40;33;01:or=01;05;37;41:mi=01;05;37
;41:ex=00;32:*.cmd=00;32:*.exe=00;32:*.com=00;32:*.btm=00;32:*.bat=00;32:*.sh=00;32:*.csh=00;32:*.tar=00;31:*
.tgz=00;31:*.arj=00;31:*.taz=00;31:*.lzh=00;31:*.zip=00;31:*.z=00;31:*.Z=00;31:*.gz=00;31:*.bz2=00;31:*.bz=00
;31:*.tz=00;31:*.rpm=00;31:*.cpio=00;31:*.jpg=00;35:*.gif=00;35:*.bmp=00;35:*.xbm=00;35:*.xpm=00;35:*.png=00;
35:*.tif=00;35:"
declare -x MAIL="/var/spool/mail/roddyduk"
declare -x OLDPWD
declare -x
PATH="/usr/kerberos/bin:/usr/local/bin:/bin:/usr/bin:/home/cis90/roddyduk/../bin:/home/cis90/roddyduk/bin:."
declare -x PWD="/home/cis90/roddyduk"
declare -x SHELL="/bin/bash"
declare -x SHLVL="2"
declare -x SSH_ASKPASS="/usr/libexec/openssh/gnome-ssh-askpass"
declare -x SSH_CLIENT="63.249.103.107 20807 22"
declare -x SSH_CONNECTION="63.249.103.107 20807 207.62.186.9 22"
declare -x SSH_TTY="/dev/pts/0"
declare -x TERM="xterm"
declare -x USER="roddyduk"
declare -x USERNAME=""
```

The env command by itself will list all the exported (environment) variables

Shell (Environment) Variables

export command – show all exported variables

To create your own environment variable use the export command

1

```
/home/cis90/roddyduk $ env | wc –l
24
/home/cis90/roddyduk $ export | wc -l
24
```

There are currently 24 environment (exported) variables

2

```
/home/cis90/roddyduk $ fan=medium
/home/cis90/roddyduk $ export fan
```

Create a new shell variable named fan and export it so it becomes an environment variable

3

```
/home/cis90/roddyduk $ env | wc -l
25
/home/cis90/roddyduk $ export | wc -l
25
```

Now there are 25 environment variables

4

```
[roddyduk@opus ~]$ export | grep fan
declare -x fan="medium"
[roddyduk@opus ~]$ env | grep fan
fan=medium
```

use grep to show the new environment variable

Flashback PS1

bash shell tip

Change PS1 to change the shell prompt

PS1 settings	Result
PS1='$PWD $'	/home/cis90/simmsben/Poems $
PS1="\w $"	~/Poems $
PS1="\W $"	Poems $
PS1="\u@\h $"	simmsben@opus $
PS1='\u@\h $PWD $'	simmsben@opus /home/cis90/simmsben/Poems $
PS1='\u@\$HOSTNAME $PWD $'	simmsben@opus.cabrillo.edu /home/cis90/simmsben/Poems $
PS1='\u \! $PWD $'	simmsben 825 /home/cis90/simmsben/Poems $
PS1="[\u@\h \W/\$"	[simmsben@opus Poems/$
PS1="Enter command: "	Enter command:

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

bash shell tip

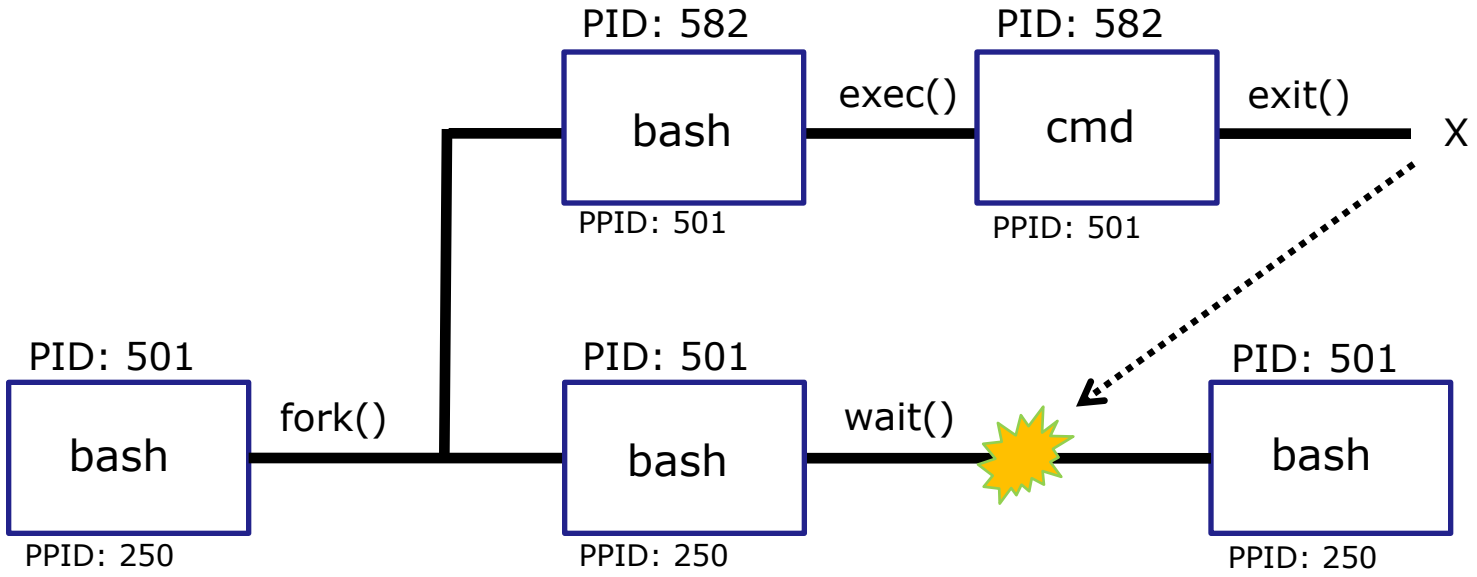
changing the prompt

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

The prompt string can have any combination of text, variables and these codes.

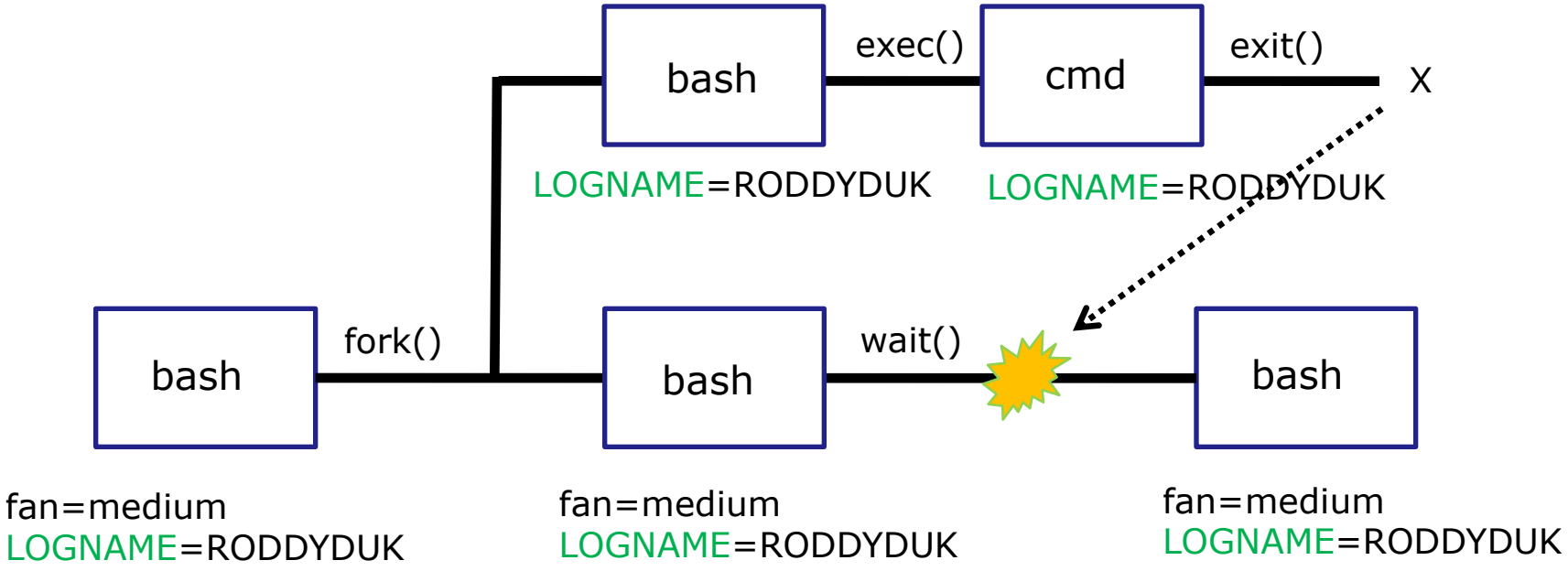
Shell Environment

exporting variables



- When a shell forks a child, not all of the variables are passed on to the child.
- Only the parent's exported variables (the environment variables) are passed to the child.

Example: Only exported variables are available to the child



fan is an ordinary shell variable, child doesn't have access to it
LOGNAME is an environment (exported) variable, child gets a copy of it

- When a shell forks a child, not all of the variables are passed on to the child.
- Only the parent's exported variables (the environment variables) are passed to the child.

Example: Only exported variables are available to the child

1

```
/home/cis90/roddyduk $ fan=medium
/home/cis90/roddyduk $ echo $fan $LOGNAME
medium roddyduk
```

fan is a shell variable
LOGNAME is an environment (exported) variable

parent

2

```
/home/cis90/roddyduk $ env | grep fan
/home/cis90/roddyduk $ set | grep fan
fan=medium
/home/cis90/roddyduk $ env | grep LOGNAME
LOGNAME=roddyduk
/home/cis90/roddyduk $ set | grep LOGNAME
LOGNAME=roddyduk
```

*fan only shows up in **set** output*

*LOGNAME shows up in both **env** and **set** output*

child

3

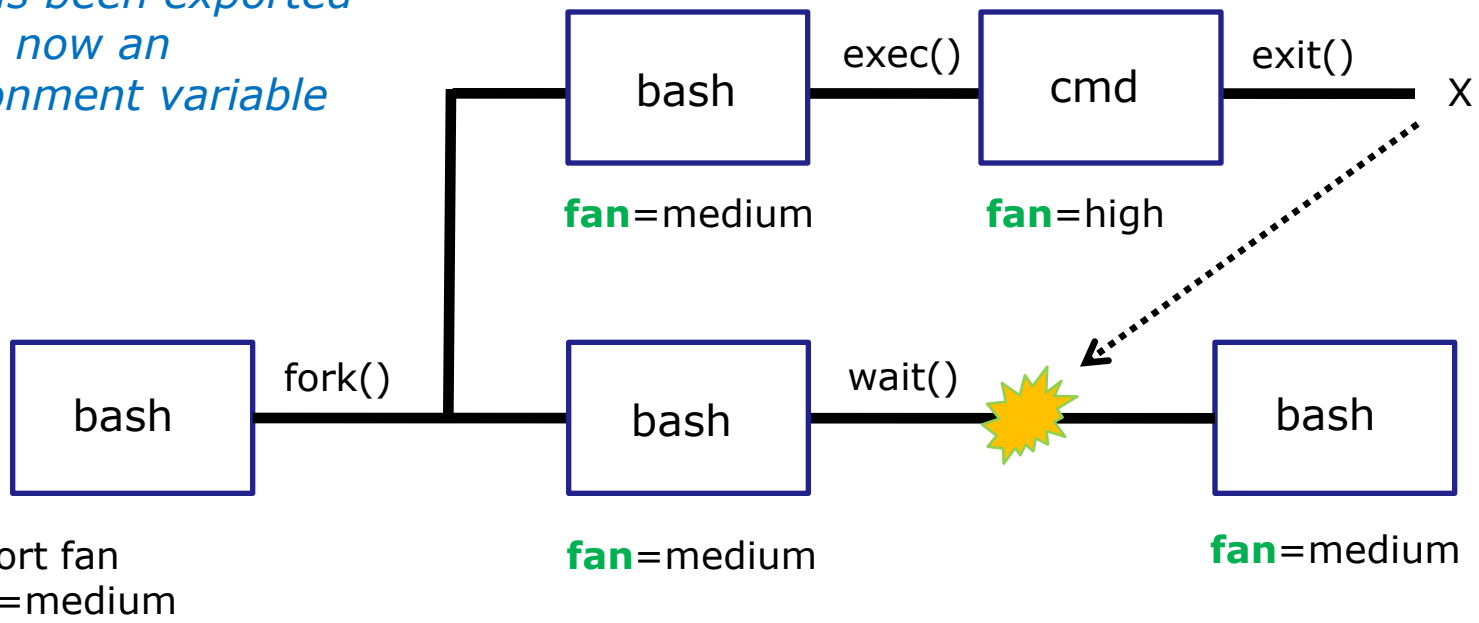
```
/home/cis90/roddyduk $ bash
[roddyduk@opus ~]$ echo $fan $LOGNAME
roddyduk
[roddyduk@opus ~]$ exit
exit
/home/cis90/roddyduk $
```

bash command runs another shell as a child process. LOGNAME is available but fan is not.

Only LOGNAME, an environment (exported) variable, is available to the child process

Example: Changes made by the child do not effect the parent

fan has been exported and is now an environment variable



- The child gets copies of the parent's environment variables which the child can change. However, any changes made by the child have no effect on the parent's variables.

The child gets a copy of the fan variable. The child can make changes to this copy but any changes made will not effect the parent's variable.

environment variables

1 parent

```
/home/cis90/roddyduk $ echo $fan
medium
/home/cis90/roddyduk $ export fan
```

export fan so it is available to children

2 child

```
/home/cis90/roddyduk $ bash
[roddyduk@opus ~]$ echo $fan
medium
```

fan is now available to the child process

3 child

```
[roddyduk@opus ~]$ fan=high
[roddyduk@opus ~]$ echo $fan
high
[roddyduk@opus ~]$ exit
exit
```

the child modifies the variable

4 parent

```
/home/cis90/roddyduk $ echo $fan
medium
```

any changes made by a child will not effect the parent's variable

aliases

alias command
(a shell builtin)

```
alias [-p] [name[=value] ...]
            Alias with no arguments or with the -p option prints the list
            of aliases in the form alias name=value on  standard  output.
            When  arguments  are  supplied,  an alias is defined for each
            name whose value is given.  A trailing space in  value causes
            the  next  word to be checked for alias substitution when the
            alias is expanded.  For each name in the  argument  list  for
            which  no  value is supplied, the name and value of the alias
            is printed.  Alias returns true unless a name  is  given  for
            which no alias has been defined.

            Note  aliases  are not expanded by default in non-interactive
            shell, and it can be enabled by  setting  the  expand_aliases
            shell option using shopt.
```

Now you can give your own name to commands!

alias command

Example: Make a new name for the cp command

1
```
/home/cis90/roddyduk $ alias copy=cp
/home/cis90/roddyduk $ copy lab09 /home/rsimms/cis90/lab09.$LOGNAME
/home/cis90/roddyduk $
```

2
```
/home/cis90/roddyduk $ type copy
copy is aliased to `cp'
/home/cis90/roddyduk $
```

*The **type** command shows that copy is an alias*

3
```
/home/cis90/roddyduk $ alias copy
alias copy='cp'
/home/cis90/roddyduk $
```

*The **alias** command (without an "=" sign) shows what the alias is*

4
```
/home/cis90/roddyduk $ unalias copy
/home/cis90/roddyduk $ alias copy
-bash: alias: copy: not found
```

*Use **unalias** command to remove an alias*

alias command

Example: Make an alias, called s, that prints the first 10 lines of smalltown

1
```
/home/cis90/roddyduk $ alias s="clear; head -10 ~/edits/small_town"
/home/cis90/roddyduk $ s
HOW SMALL IS SMALL?

YOU KNOW WHEN YOU'RE IN A SMALL TOWN WHEN...
   The airport runaway is terraced.
   The polka is more popular than a mashpit on on Saturday night.
   Third Street is on the edge of town.
   Every sport is played on dirt.
   The editor and publisher of the newspaper carries a camera at all times.
   You don't use your turn signal because everyone knows where you are
      going knows where you are going.
/home/cis90/roddyduk $
```

2
```
/home/cis90/roddyduk $ type s
s is aliased to `clear; head -10 ~/edits/small_town'
/home/cis90/roddyduk $ alias s
alias s='clear; head -10 ~/edits/small_town'
```

*The **type** and **alias** commands show that s is an alias*

3
```
/home/cis90/roddyduk $ unalias s
/home/cis90/roddyduk $
```

*Use **unalias** command to remove an alias*

alias an alias

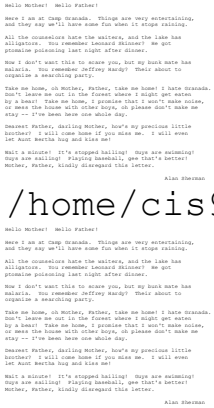
Yes, an alias can be made using another alias

1

```
/home/cis90/roddyduk $ alias show=cat
/home/cis90/roddyduk $ alias view=show
```

*Make **show** an alias of **cat***
*Make **view** and alias of **show***

```
/home/cis90/roddyduk $ show letter
```



reduced sized to fit on page

2

*Now, either **show letter** or **view letter** will cat out the letter file*

```
/home/cis90/roddyduk $ view letter
```


reduced sized to fit on page

3

```
/home/cis90/roddyduk $ unalias show
/home/cis90/roddyduk $ alias view
alias view='show'
/home/cis90/roddyduk $ view letter
-bash: show: command not found
/home/cis90/roddyduk $
```

It can be broken too

single and double quotes (very subtle)

You can control whether bash does filename expansion when you create the alias or ... when the alias is used

```
$ ac=on
$ fan=medium
$ defrost=off
```

double → *single*

1

```
$ alias p="echo $ac $fan $defrost"
$ alias p
alias p='echo on medium off'
```

```
$ alias p='echo $ac $fan $defrost'
$ alias p
alias p='echo $ac $fan $defrost'
```

2

```
$ p
on medium off
```

```
$ p
on medium off
```

3

```
$ ac=off
```

```
$ ac=off
```

4

```
$ p
on medium off
```

```
$ p
off medium off
```

Note: using single quotes prevents bash from expanding the variables when creating up the alias

Class Exercise

Make some aliases

For example:

- **alias mypath="echo $PATH"**
- **mypath**

- **alias details=file**
- **details /usr/bin/spell**

Now invent 2-3 of your own

bash startup files

bash startup files

only executed when logging in

/etc/profile (system wide)

- adds root's special path

/etc/profile.d/*.sh (system wide)

- kerberos directories added to path
- adds color, vi aliases
- language, character sets

.bash_profile (user specific)

- set up your path, prompt and other environement variables

.bashrc (user specific)

- add your new aliases here

Edit these files to customize your shell environment

/etc/bashrc (system wide)

- changes umask to 0002 for regular users
- sets final prompt string

.bash_profile

.bash_profile

- The .bash_profile is a shell script that sets up a user's shell environment.
- This script is executed each time the user logs in.
- The .bash_profile is used for initializing shell variables and running basic commands like umask or set -o options.
- This script also runs the users .bashrc file

.bash_profile (runs only at login)

```
[roddyduk@opus ~]$ cat .bash_profile
# .bash_profile

# Get the aliases and functions
if [ -f ~/.bashrc ]; then
        . ~/.bashrc
fi

# User specific environment and startup programs

PATH=$PATH:$HOME/../bin:$HOME/bin:.
BASH_ENV=$HOME/.bashrc
USERNAME=""
PS1='$PWD $ '
export USERNAME BASH_ENV PATH
umask 002
set -o ignoreeof
stty susp ^F
eval `tset -s -m vt100:vt100 -m :\?${TERM:-ansi} -r -Q `

[roddyduk@opus ~]$
```

sources the .bashrc file

Appends the user's bin and the "current directory" to the path

These variables are set

Prompt (PS1) used in CIS 90 is specified

These variables are exported

umask value is set

EOF's are ignored

Suspend character redefined from Z to F

Terminal type is set

.bashrc

.bashrc

- The .bashrc is a shell script that is executed during user login and whenever a new shell is invoked
- Good place to add user defined aliases

.bashrc

The .bashrc is a shell script that is executed during user login and whenever a new shell is invoked. This file usually contains the user defined aliases. e.g.

```
[roddyduk@opus ~]$ cat .bashrc
# .bashrc

# User specific aliases and functions

# Source global definitions
if [ -f /etc/bashrc ]; then
        . /etc/bashrc
fi
alias print="echo -e"
[roddyduk@opus ~]$
```

sources the /etc/bashrc file

creates a print alias, the -e option enables interpretation of backslash escapes

Class Exercise

Modify .bashrc

Add a new permanent alias to your bash environment

```
alias me="finger $LOGNAME"
```

When finished logout and login again and verify the alias is permanent.

. and exec

. and exec

In normal execution of a unix command, shell-script or binary, the child process in unable to affect the login shell environment.

Sometimes it is desireable to run a shell script that will initialize or change shell variables in the parent environment. To do this, the shell (bash) provides a **.** (dot) or **source** command, which instructs the shell to execute the shell script itself, without spawning a child process to run the script, and then continue on where it left off.

. *myscript*
source *myscript* } *equivalent*

In this example, the commands in the file script are run by the parent shell, and therefore, any changes made to the environment will last for the duration of the login session.

If a UNIX command is run using the exec command, the bash code in the process is replaced by the command code, when finished the process will terminate when that command exits

exec clear

This will have the effect of clearing the screen and logging off the computer

Wrap up

Lab 10 - the last one!

Cabrillo College

CIS 90 Linux Lab Exercise

Lab 10: The Shell Environment

Spring 2012

Lab 10: The Shell Environment

In this lab you will customize your login environment to suit your needs and preferences. By modifying environment variables and editing your *.bash_profile* and *.bashrc* files, you will customize your shell environment in a number of different ways.

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. Start this lab from your home directory.

Environment Variables

1. Display the contents of your PWD environment variable. Change to your *bin* subdirectory and display the same variable. How did it change?
2. Change back to your home directory.
3. Display the contents of your PATH environment variable. Note the colon (:) separating the different directory names. What is the last directory in which the system searches for commands?
4. Make a new environment variable called GREETING and assign it an appropriate salutation. Don't forget to use quotes if your message has whitespace in it.
5. Use the **env** command to see if it is in your environment. Is it there? What must you do to put it in the environment?
6. Export the variable GREETING and use **env** to verify it's there.
7. Invoke a new bash shell process by typing:
   bash
   Now use the **unset** command to unset the variable PS1. What Happened?
8. Reset the PS1 variable by entering the following command:
   PS1="Yes master: "
   What happens to your primary prompt?

Extra Credit Special

1) Why did the prompt change?

```
/home/cis90/roddyduk $ bash
[roddyduk@opus ~]$ exit
exit
/home/cis90/roddyduk $
```

2) What command could be issued prior to the bash command above that would prevent the prompt from changing?

For 5 points extra credit, email risimms@cabrillo.edu your complete answer by noon tomorrow (May 3, 2012)

New commands:

.	- source the commands
alias	- create or show an alias
unalias	- remove an alias
set	- show all variables
env	- show environment variables
export	- export variable so child can use
exec	- replace with new code
source	- same as .

New Files and Directories:

.bash_profile	- executed at login
.bashrc	- executed at login and new shells

Next Class

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

Lab 10

Quiz questions for next class:

- How do you make an alias setting permanent?
- What must you do to a variable so a child can use it?
- How would you use an alias to make a command named copy … that would do what the cp command does?

Backup

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?

```
/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 ???????????

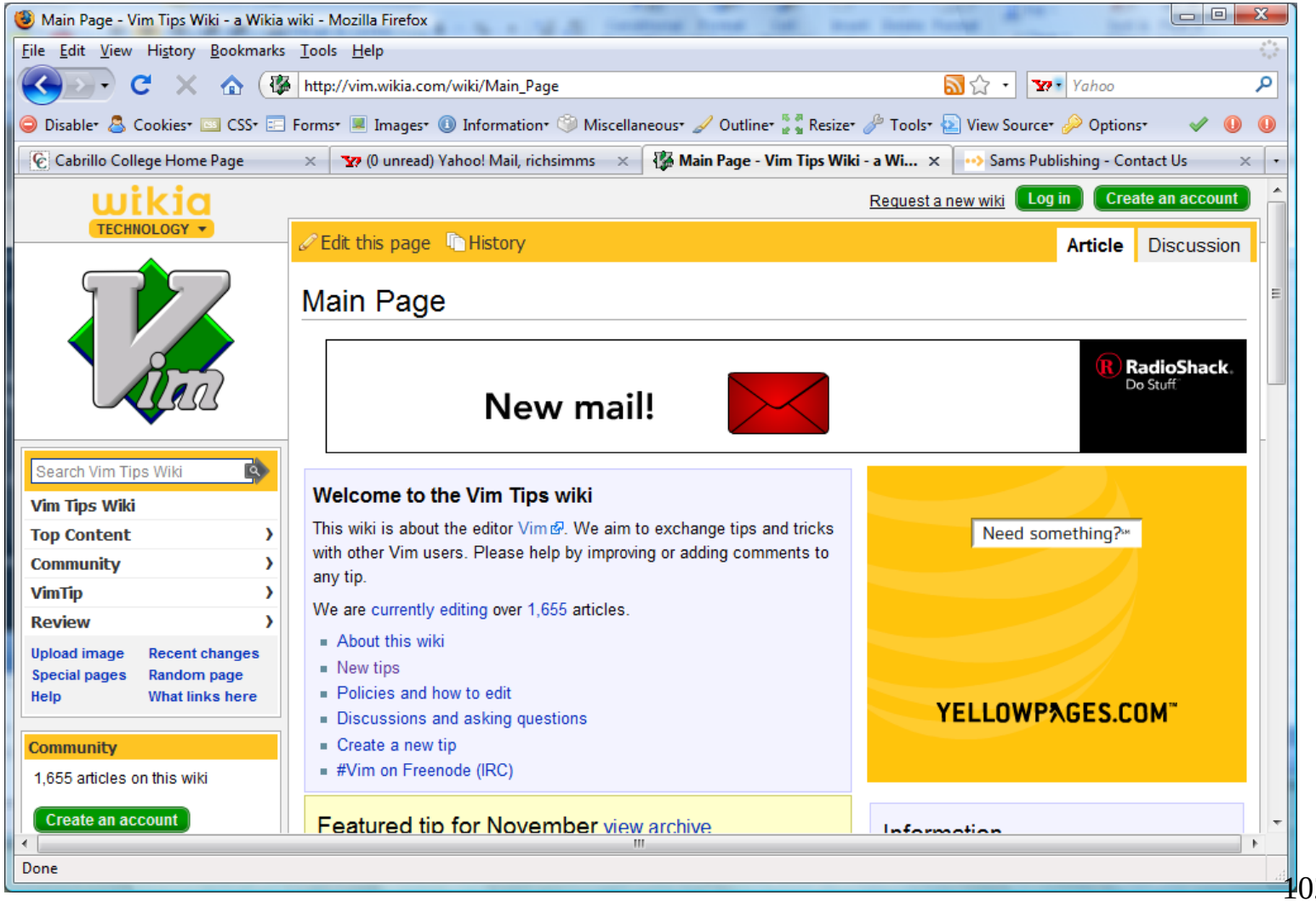
*OK, the actual command is **aspell***

aspell command

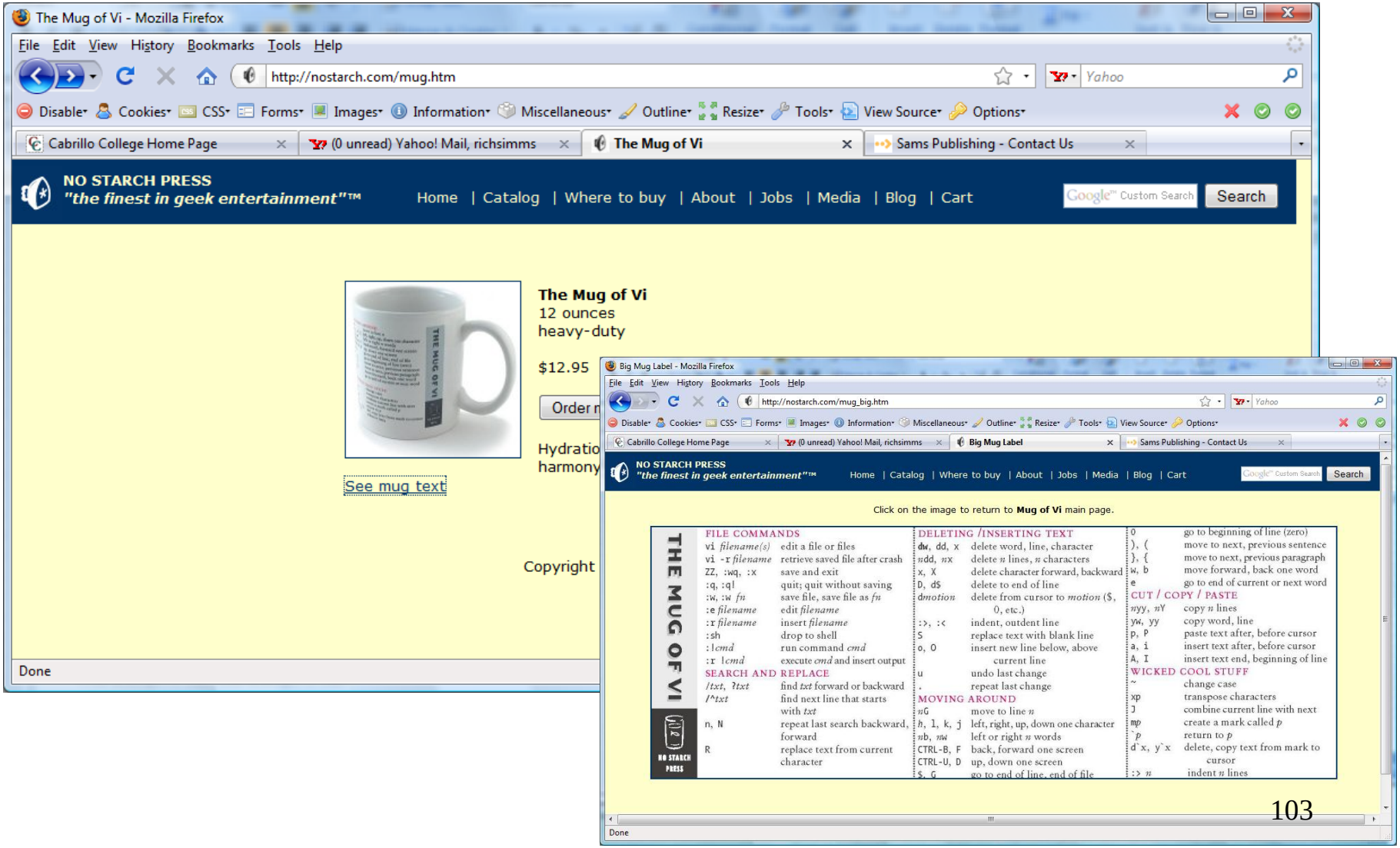
```
ASPELL(1)                  Aspell Abbreviated User's Manual                  ASPELL(1)

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

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



The Mug of vi


The Mug of Vi - Mozilla Firefox
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
The Mug of Vi
12 ounces
heavy-duty
$12.95
See mug text
Copyright
Done
Big Mug Label - Mozilla Firefox
http://nostarch.com/mug_big.htm
Click on the image to return to Mug of Vi main page.
THE MUG OF VI
FILE COMMANDS
DELETING /INSERTING TEXT
CUT / COPY / PASTE
SEARCH AND REPLACE
MOVING AROUND
WICKED COOL STUFF
NO STARCH PRESS