

Lesson Module Status

- Slides – draft
 - Properties - done
 - Flash cards –
 - First minute quiz – done
 - Web calendar summary – done
 - Web book pages – done
 - Commands – done
 - Lab – done
 - Supplies () - na
 - Class PC's – na
 - Chocolates - bringing
-
- Backup headset charged - done
 - CCC Confer wall paper - done
 - Slides & Lab uploaded - done
 - Final project posted - done
 - Extra credit lab posted - done



Dennis



Sean



Christopher



Francisco



Rich

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



Salena



Abd



Sarah



Astitow



Mike D.



Alex



Christine



Steven



Richie



Nathan



Tony



James G.



Sergio



Anthony



Fernando



Miguel



Lars



Jennifer



Rudy



Laura P.



Nick



Juan



Jacob



Andrew



Luke

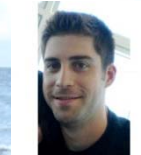


Saulius

Online Class Students



Edtson



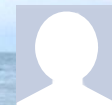
James B.



Liz



Casady



Jason



Aaron



Steve



Songul



Stephanie



Victor



Tanya



Mike P.



Adriana



Laura S.



Olivia



Janelle

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

Quiz

Please close your books, notes, lesson materials, forum and answer these questions **in the order** shown:

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

email answers to: risimms@cabrillo.edu

(If you are in the classroom you can write your answers on a scrap piece of paper and hand it in)



- [] Has the phone bridge been added?
- [] Is recording on?
- [] Does the phone bridge have the mike?
- [] Share slides, putty (rsimms, simmsben, roddyduk), and Chrome
- [] Disable spelling on PowerPoint

The Shell Environment

Objectives

- 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.

Agenda

- Quiz
- Housekeeping
- Spell checking
- vi and /bin/mail
- Review pathnames
- Final project prep
- Variables
- The shell environment
- Aliases
- .bash_profile
- .bashrc



Housekeeping

Previous material and assignment

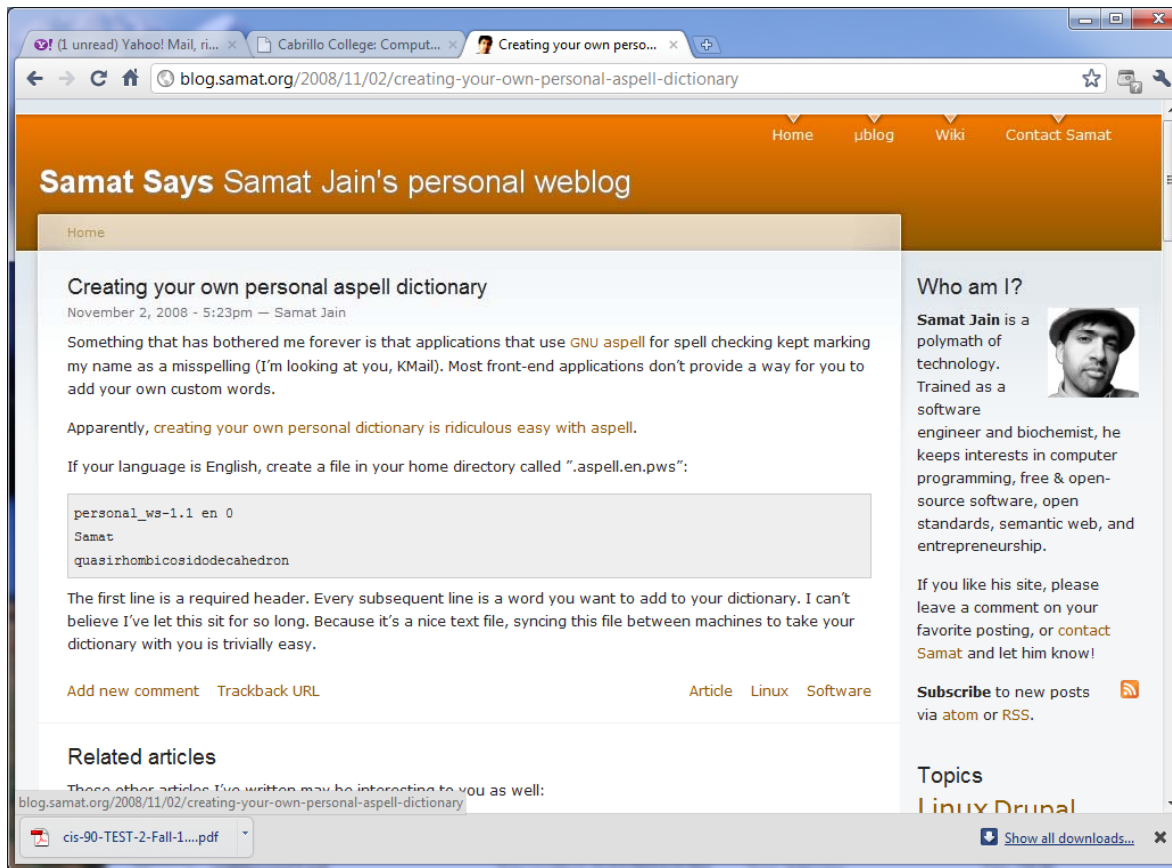
1. Lab 9 due midnight tonight
2. Five posts due midnight tonight

Reminder: Only posts between in the CIS 90 forum between 10/21 and 11/17 (inclusive) are be counted.

3. Questions?
 - vi
 - lab 9
 - test 2
 - pathnames

Ayshire mashpit and personal dictionaries

aspell command



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

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

Make a Personal Dictionary

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

```
cd edits/  
spell small_town
```

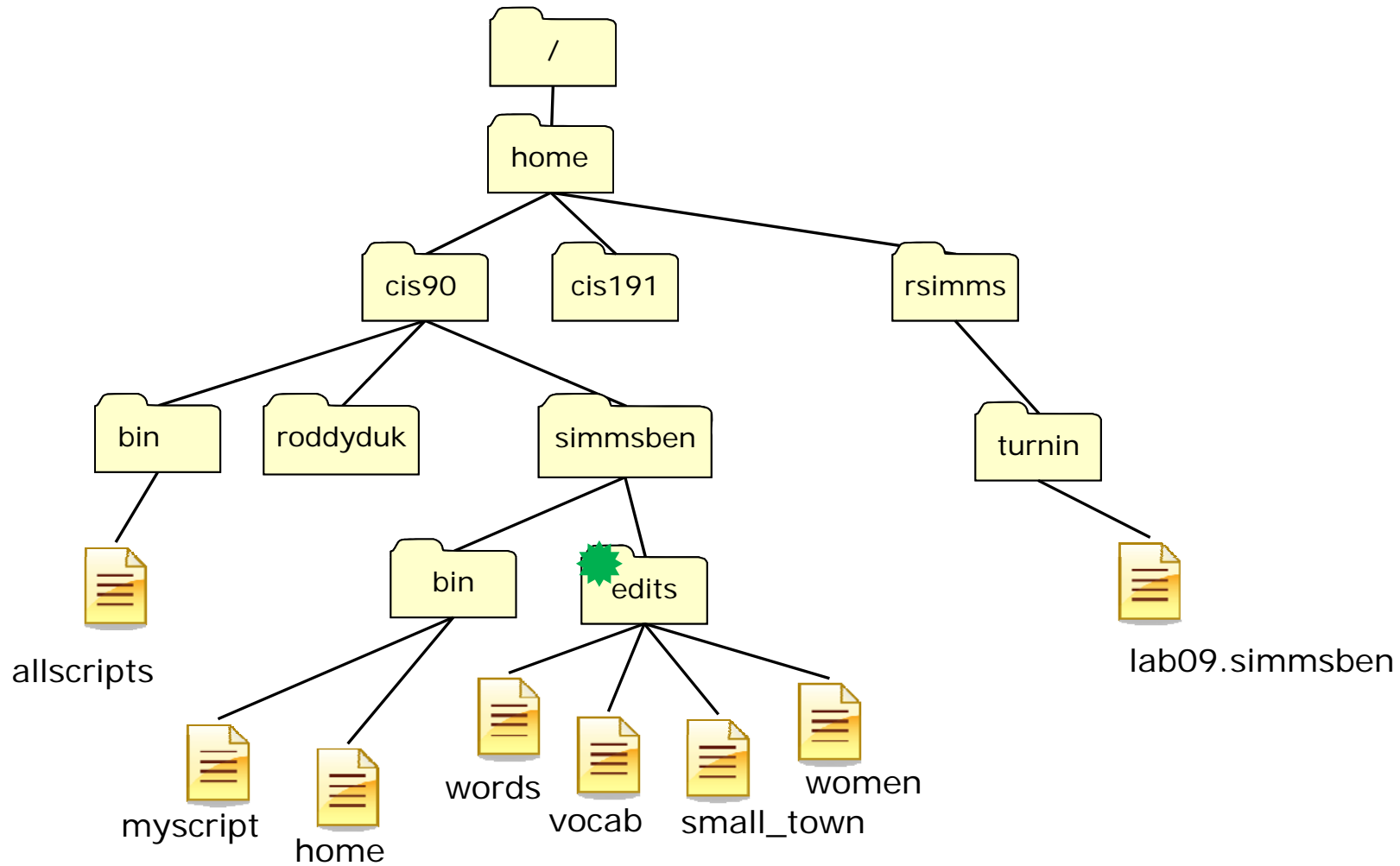
Note: You should still leave the two words Ayshire and mashpit in the file words when you submit Lab 9



pathnames

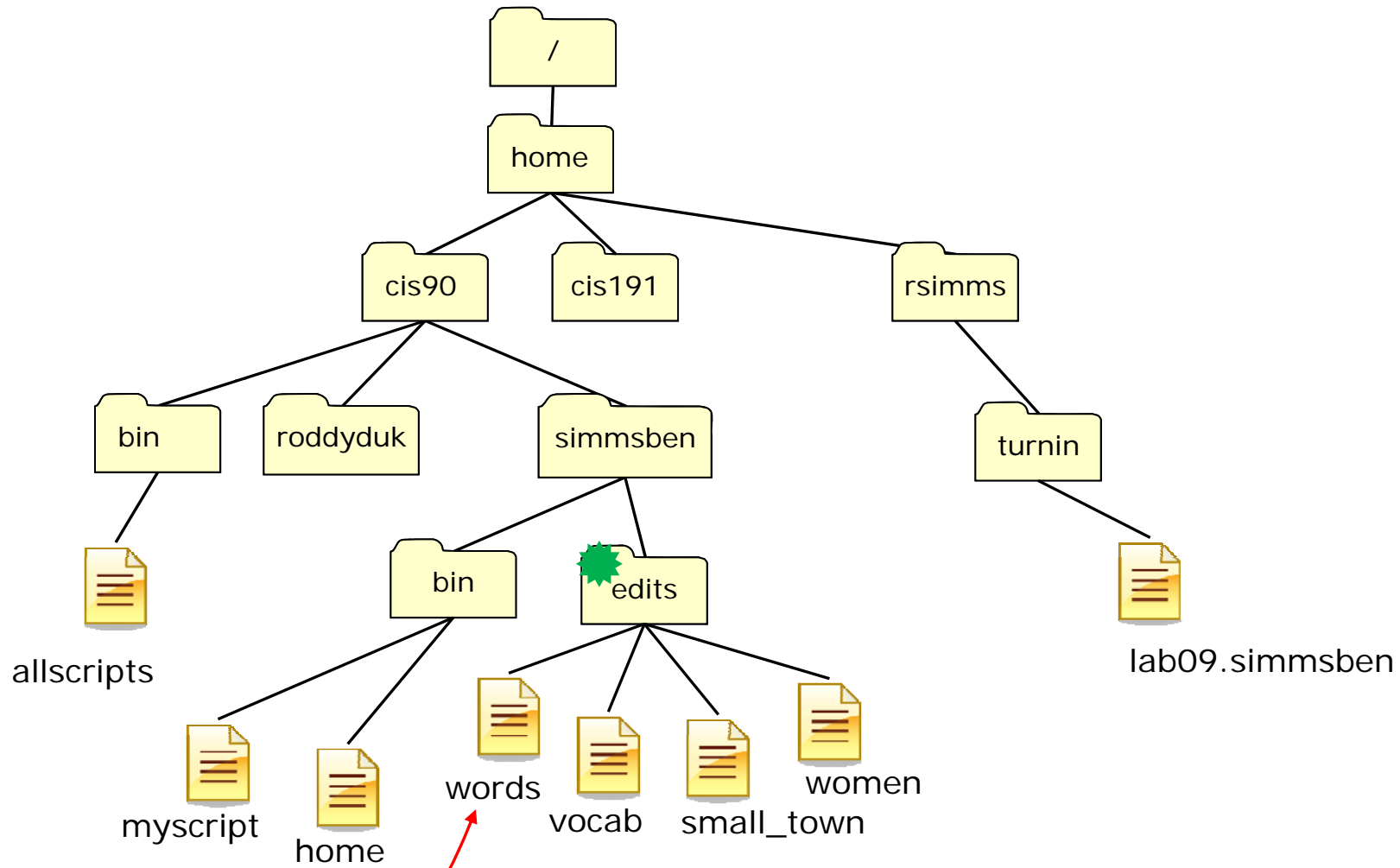
Reminder

- You must always use complete pathnames when specifying files as arguments on a command.
- Pathnames can be relative or absolute.



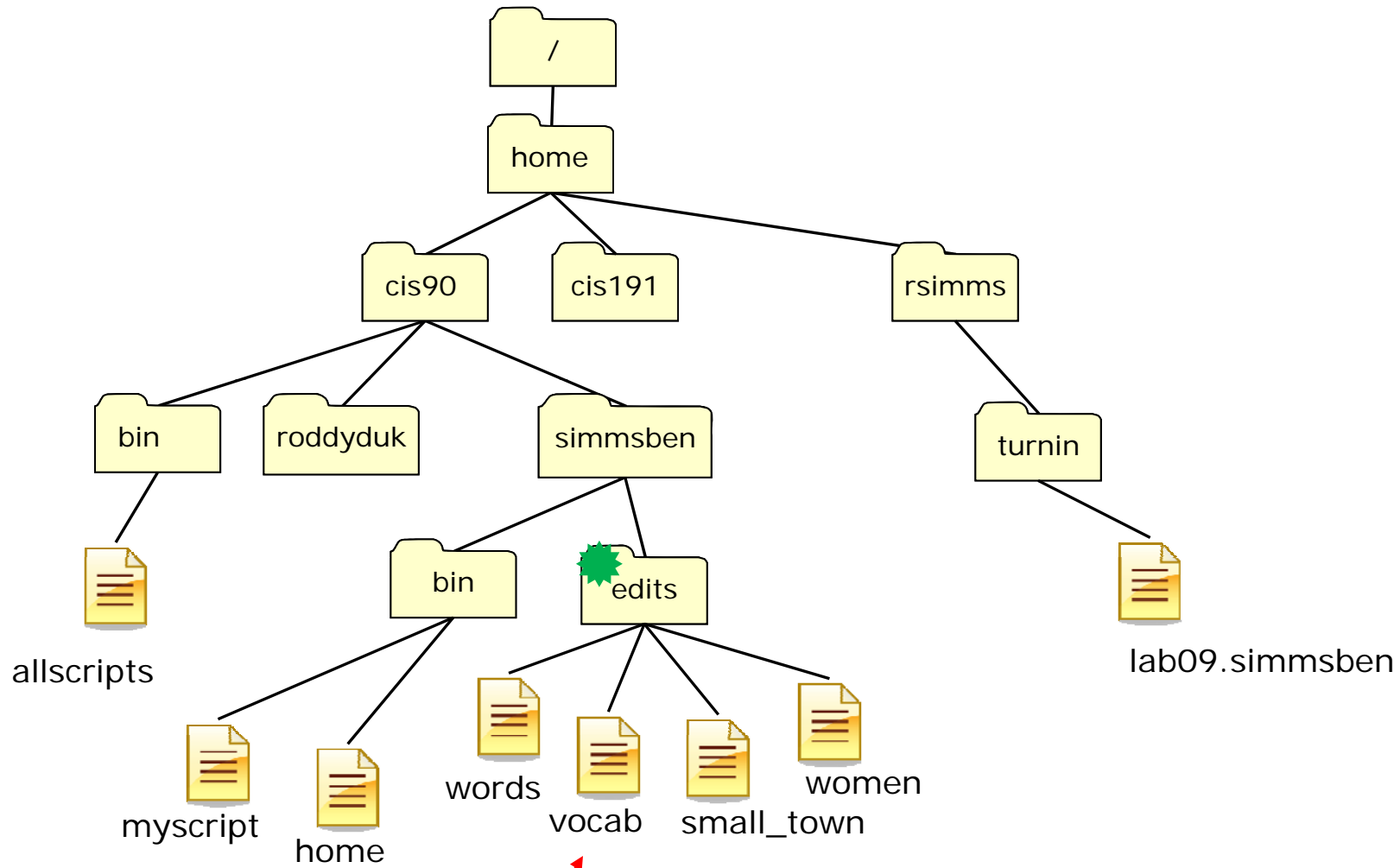
cat `../bin/home` words vocab small_town woman > /home/rsimms/turnin/lab09.\$LOGNAME

relative pathname



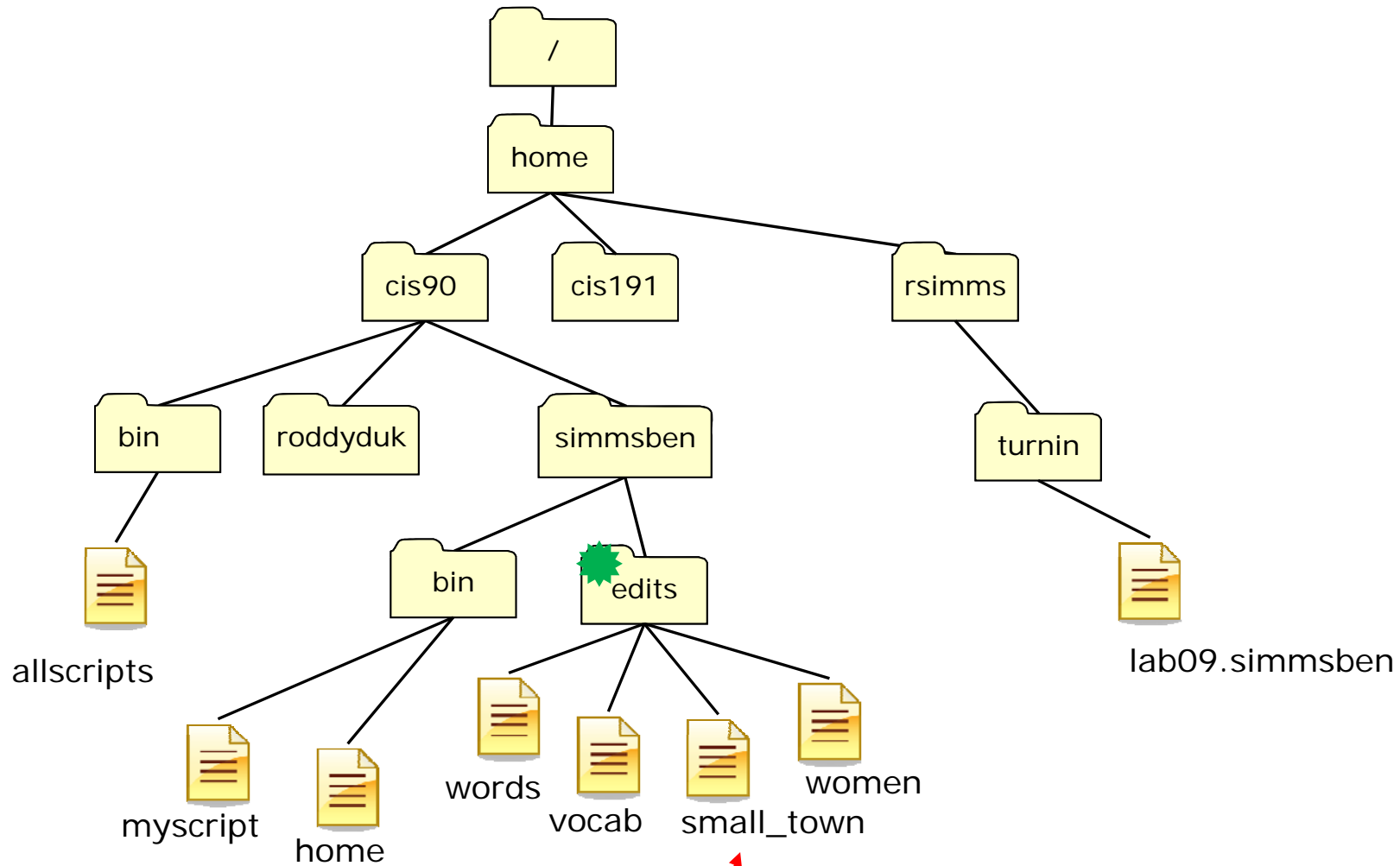
`cat ../bin/home words vocab small_town woman > /home/rsimms/turnin/lab09.$LOGNAME`

relative pathname



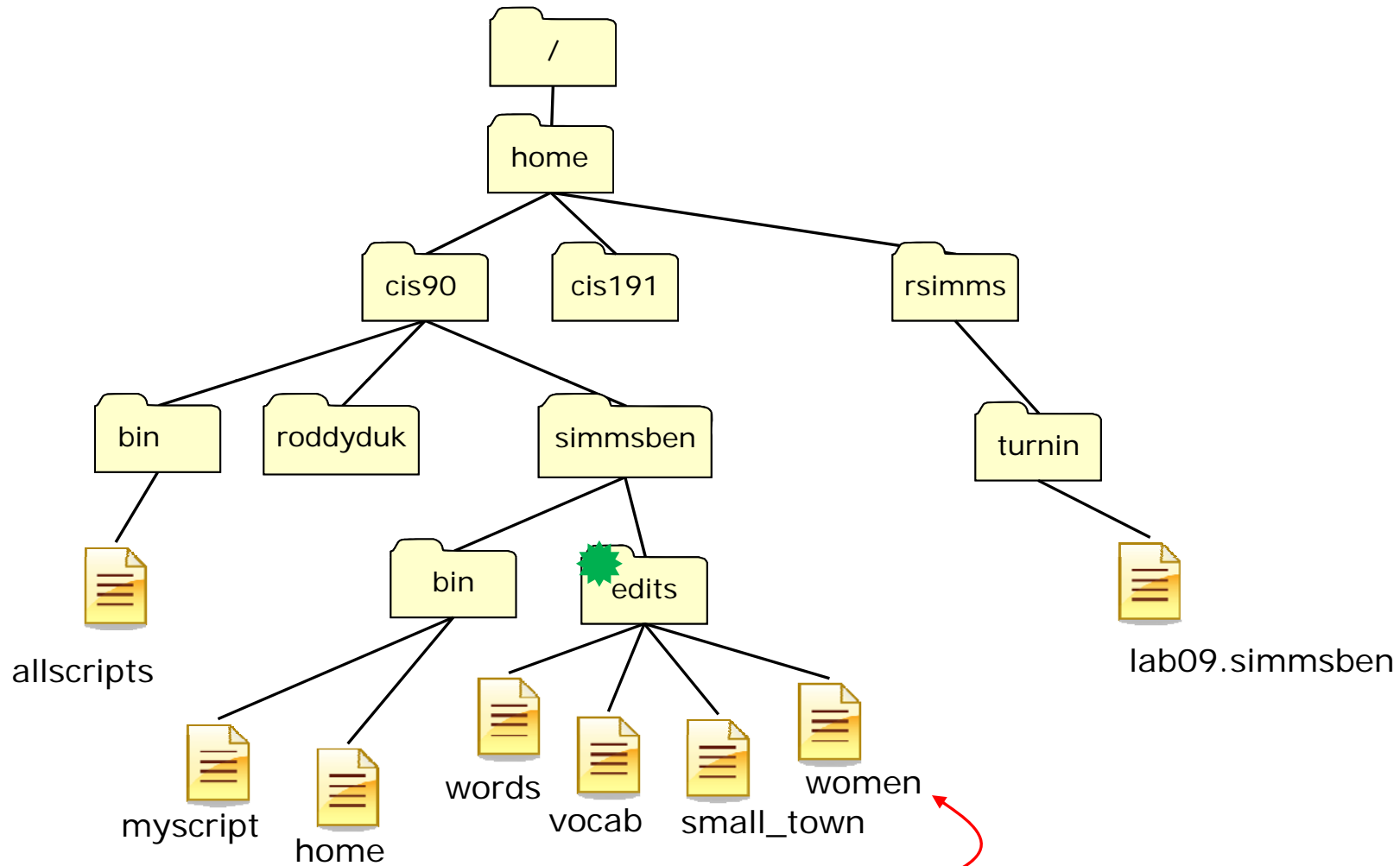
`cat ../bin/home words vocab small_town woman > /home/rsimms/turnin/lab09.$LOGNAME`

relative pathname



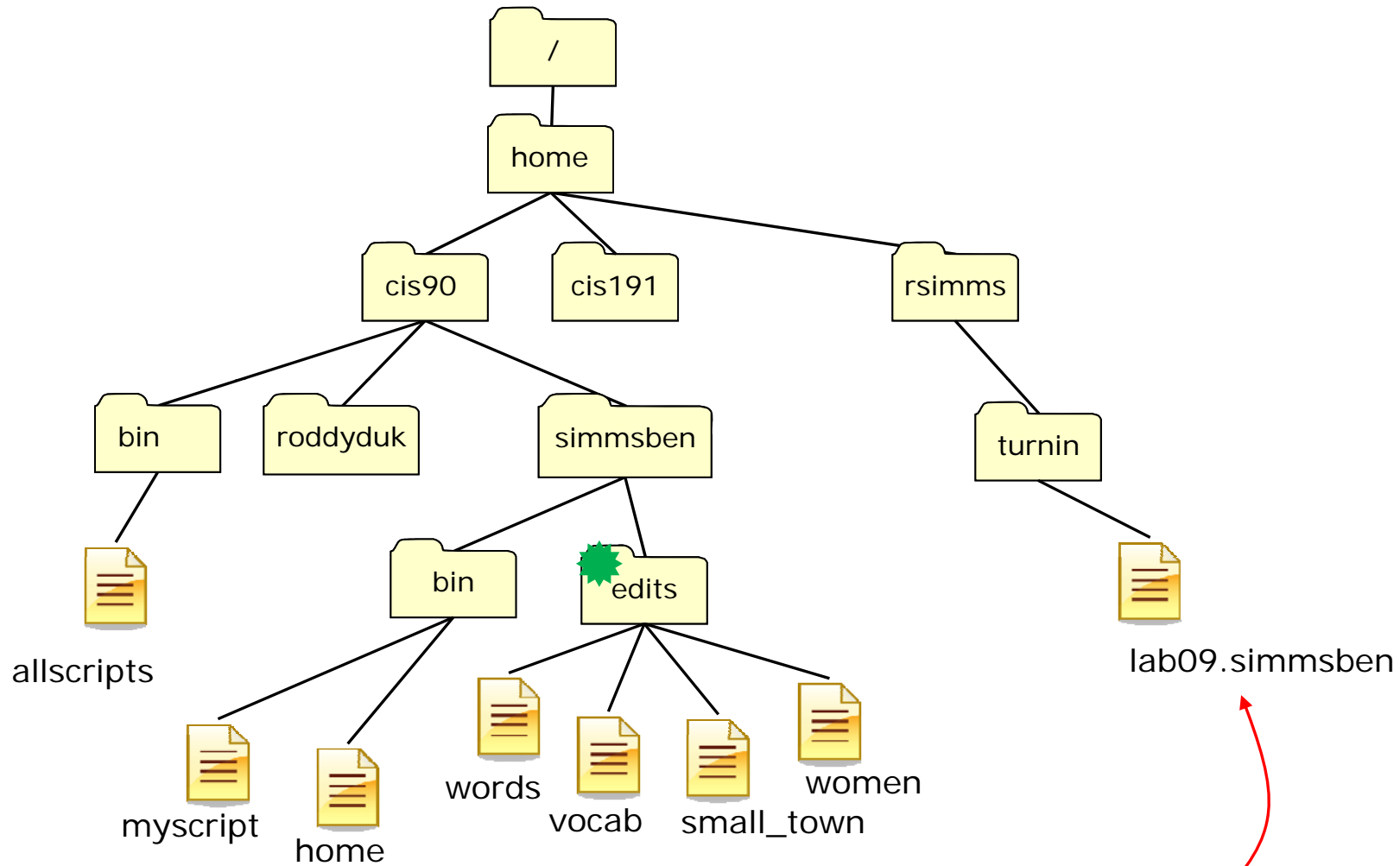
```
cat ../bin/home words vocab small_town woman > /home/rsimms/turnin/lab09.$LOGNAME
```

relative pathname



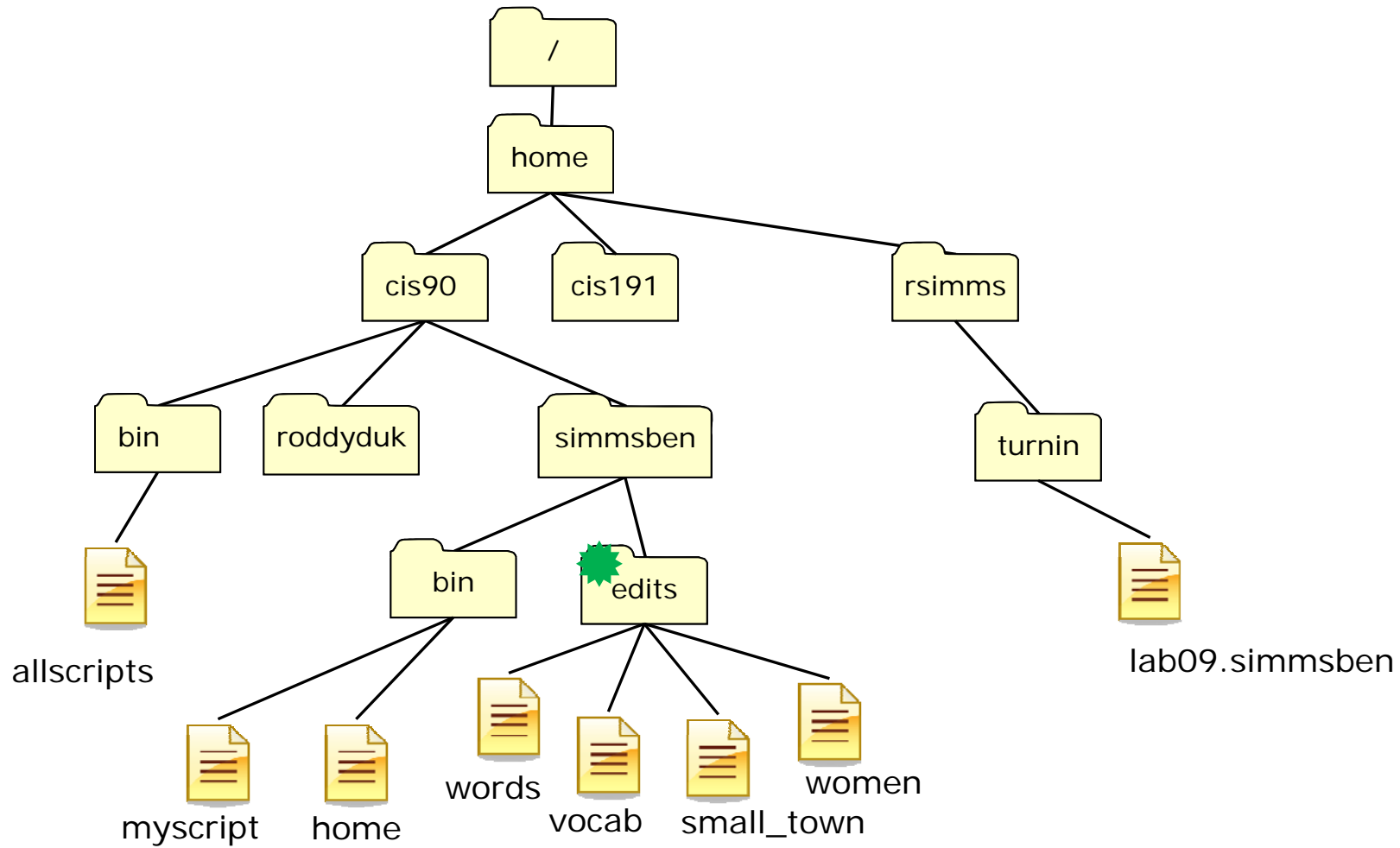
`cat ../bin/home words vocab small_town woman > /home/rsimms/turnin/lab09.$LOGNAME`

relative pathname



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

absolute pathname



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

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

```
/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.cabrillo.edu 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!

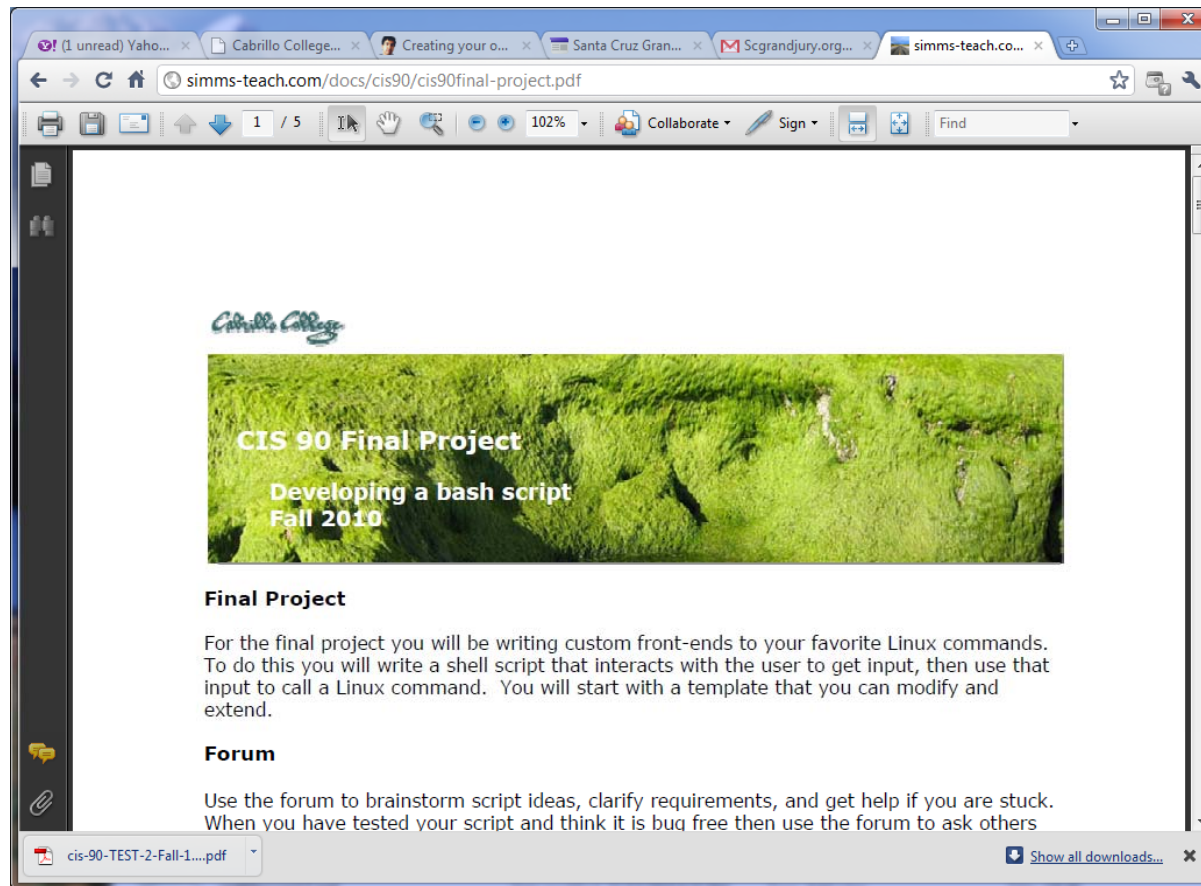
Use /bin/mail and send me a message that you have made or corrected using the ~vi command

cc: yourself so you can verify what you sent.



final project

Final Project

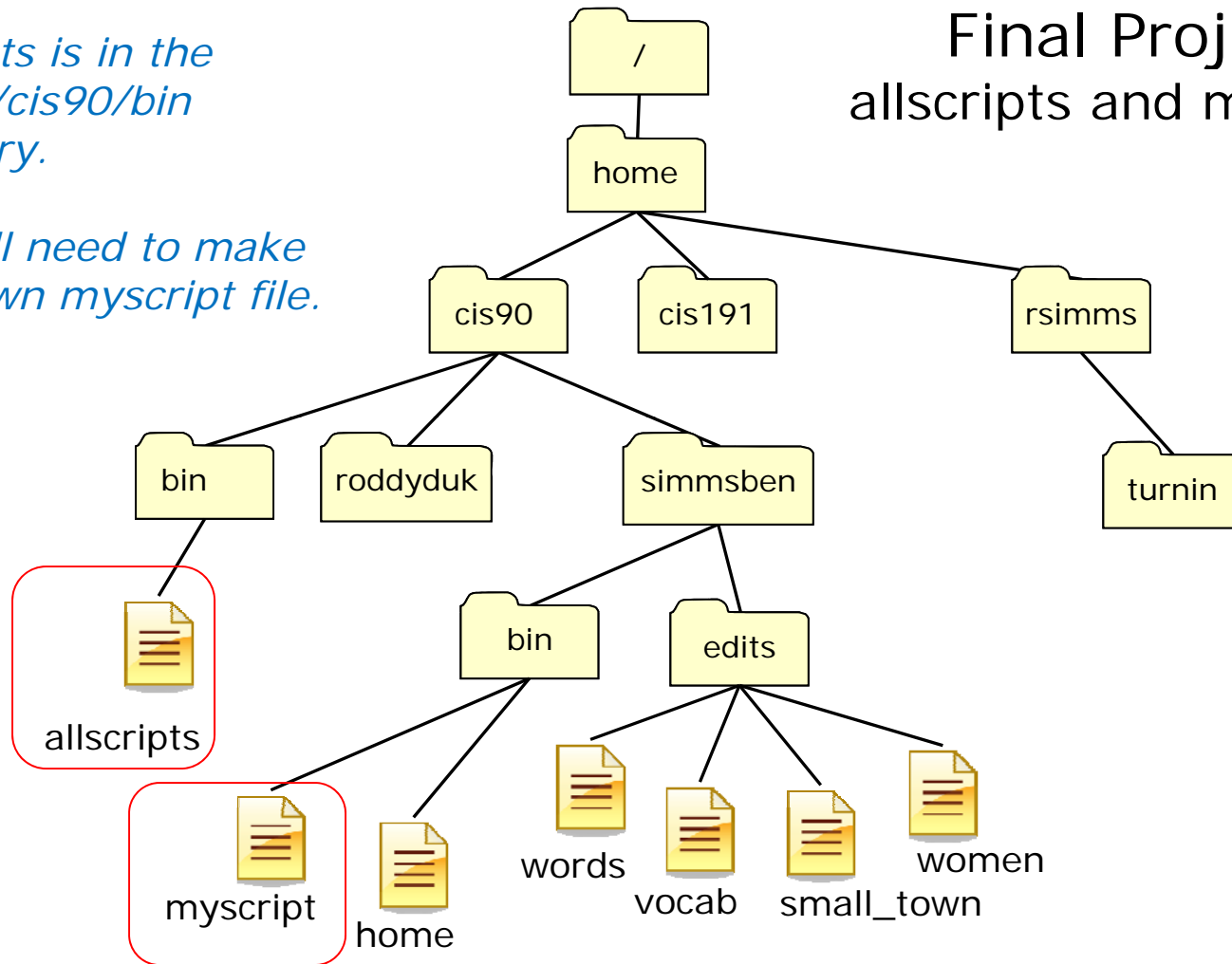


You now have the necessary skills to begin the final project!

Final Project allscripts and myscript

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

*You will need to make
your own myscript file.*



```

/home/cis90/rododyduk $ ls -l /home/cis90/bin/allscripts bin/myscript
-rwxr-xr-x 1 rododyduk 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 "
*****
*                Fall 2010 CIS 90 Projects                *
*****
1) Aaron          13) Elisabeth   25) Laura S.   37) Saulius
```

Print a menu with each student's name

More menu lines (reduced is size so this fits on one page)

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

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

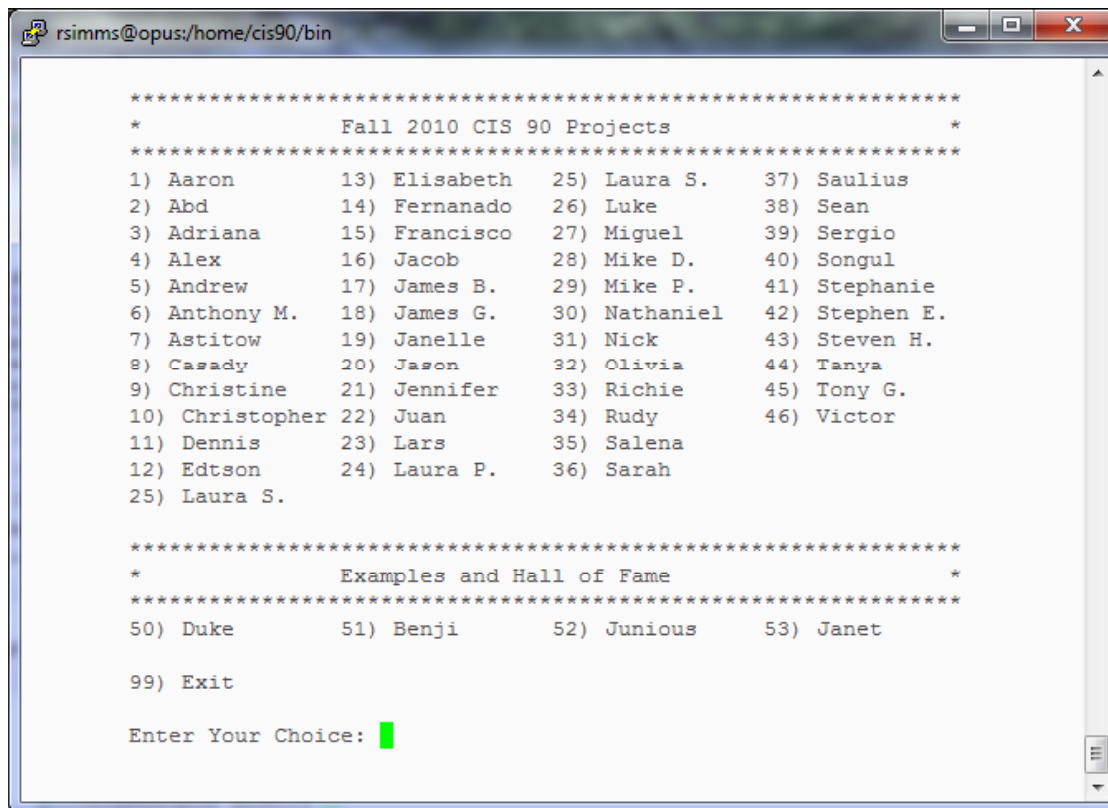
The rest of the pathnames to student's myscript files (reduced is size so this fits on one page)

```
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



```
rsimms@opus:/home/cis90/bin
*****
*                               *
*           Fall 2010 CIS 90 Projects           *
*                               *
*****
1) Aaron          13) Elisabeth  25) Laura S.   37) Saulius
2) Abd            14) Fernanado  26) Luke      38) Sean
3) Adriana       15) Francisco  27) Miguel    39) Sergio
4) Alex          16) Jacob      28) Mike D.   40) Songul
5) Andrew        17) James B.   29) Mike P.   41) Stephanie
6) Anthony M.    18) James G.   30) Nathaniel 42) Stephen E.
7) Astitow       19) Janelle    31) Nick       43) Steven H.
8) Casady        20) Jason      32) Olivia     44) Tanya
9) Christine     21) Jennifer   33) Richie     45) Tony G.
10) Christopher  22) Juan       34) Rudy       46) Victor
11) Dennis       23) Lars       35) Salena
12) Edtson       24) Laura P.   36) Sarah
25) Laura S.

*****
*                               *
*           Examples and Hall of Fame           *
*                               *
*****
50) Duke          51) Benji      52) Junious    53) Janet

99) Exit

Enter Your Choice: █
```

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

```

#!/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

```

Your initial myscript file will look like this in vi

vi understands shell scripts and will use color 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

```
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"
            ;;
    esac
done
```

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

```
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"
        ;;
```

A variable (\$ means "the value of")

another variable

another variable

Variables (\$ means "the value of")

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"
            ;;
    esac
done
```

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 are names consisting 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:
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 the value from 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 **\$_____** 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]\$
```

Shell Variables

Changing values of variables

Use = (no spaces) to change value

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

Shell Variables

```

SHELL          LOGNAME      HOME          LANG
SSH_TTY        EUID          PWD
BASH_VERSION   LINES        COLORS        PPID
               IFS
               BASH_ENV    HOSTNAME
MAILCHECK      SHELLOPTS
USER           BASH         PS4           TERM          PIPESTATUS   GROUPS
HISTFILESIZE   OPTIND
               UID          BASH_VERSINFO
BASH_ARGV      PATH         PS1
SHLVL          tmpid        SSH_CONNECTION
               USERNAME    OSTYPE       HISTFILE
BASH_ARGC     USERNAME    OSTYPE       HISTFILE
HISTSIZ      BASH_LINENO LESSOPEN
               OPTERR
HOSTTYPE      LS_COLORS   CVS_RSH
COLUMNS     INPUTRC     BASH_SOURCE  _            MACHTYPE
PROMPT_COMMAND
DIRSTACK     MAIL        SSH_ASKPASS  G_BROKEN_FILENAMES

```

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
HISTSIZ=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 by itself will show all the 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

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

create some new shell variables

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

*print the values of
the shell variables*

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

*print the names of
the shell variables*

Class Exercise

Create and initialize 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 remove some new shell variables
```

```
/home/cis90/roddyduk $ echo $defrost $ac $fan Non-existing variables  
have null values
```

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

Class Exercise

Create and initialize 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

```

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

```

BASH=/bin/bash
BASH_ARGC=( )
BASH_ARGV=( )
BASH_SOURCE={ }
BASH_ENV=/home/cis90/roddyduk/.bashrc
BASH_LINENO=( )
BASH_REMATCH=( )
BASH_VERSION={ [0]*3* [1]*2* [2]*25* [3]*1* [4]*release* [5]*1686-redhat-linux-gnu* }
BASH_VERSINFO=( 3.2.2611-release
COLORS=/etc/dircolors.xterm
COLLINES=84
CYS_RSH=ssh
DIRSTACK=( )
EUID=1156
GID=501
GROUPE=( )
G_BROKEN_FILENAMES=1
HISTFILE=/home/cis90/roddyduk/.bash_history
HISTFILESIZE=1000
HISTSIZE=1000
HOME=/home/cis90/roddyduk
HOSTNAME=opus.cabrillo.edu
HOSTTYPE=i686
IFS=$'\n\t'
JOBNAME=10
LANG=C
LINES=19
LESSOPEN=| /usr/bin/lesspipe.sh %s
LINES=19
LINUX=roddyduk
LC_COLORS="rs=0:fi=0:di=0:34:ln=0:36:pi=40:33:ao=0:35:bd=40:33:01:or=01:05:37:41:mi=01:05:37:41:ex=00:32*:cm=00:32*:ex=00:32*:co=00:32*:bc=00:32*:bat=00:32*:sh=00:32*:csh=00:32*:tar=00:31*:tgz=00:31*:arj=00:31*:taz=00:31*:lzh=00:31*:zip=00:31*:z=00:31*:d=00:31*:g=00:31*:b2=00:31*:bz=00:31*:t=00:31*:rpm=00:31*:cpio=00:31*:gif=00:35*:lmp=00:35*:x86=00:35*:xps=00:35*:png=00:35*:tif=00:35*"
MAIL=/var/spool/mail/roddyduk
MAILCHECK=60
OLDPWD=/home/cis90/roddyduk/edite
OPTERR=1
OPTIND=1
OSTYPE=linux-gnu
PATH=/usr/sbin:/bin:/usr/local/bin:/usr/bin:/home/cis90/roddyduk/.../bin:/home/cis90/roddyduk/bin
PIPESTATUS={ [0]*G* }
PPID=7254
PROMPT_COMMAND="echo -ne '\033[0;${USER}@${HOSTNAME%%.*}:[PND]${HOME/~}': echo -ne '\007'"
PS1=" $ "
PS2=" "
PS4=" + "
PWD=/home/cis90/roddyduk
SHELL=/bin/bash
SHELLOPTS=braceexpand:emacs:hashall:histexpand:ignoreeof:interactive-comments:monitor
SHLVL=1
SSH_ASKPASS=/usr/libexec/openssh/gnome-ssh-askpass
SSH_CLIENT="63.249.103.107 19509 22"
SSH_CONNECTION="63.249.103.107 19509 207.62.186.9 22"
SSH_TTY=/dev/pts/1
TERM=eterm
UID=1156
USER=roddyduk
USERRC=
_

```

font reduced for the other variables to fit on slide

```

ac=off
defrost=on
fan=medium

```

Shell Variables

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

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

Shell Variables

```
/home/cis90/roddyduk $ vi input_ac  
/home/cis90/roddyduk $ cat input_ac
```

```
#!/bin/bash
```

```
echo -n "Turn the Air Conditioning on or off? " Using a variable in a script
```

```
read ac
```

```
echo "Air Conditioning set to $ac"
```

```
exit
```

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

Add execute permissions

```
/home/cis90/roddyduk $ ./input_ac
```

```
Turn the Air Conditioning on or off? off
```

```
Air Conditioning set to off
```

Run script

Class Exercise

Now try the dialog script:

vi input_ac

add 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 input_ac

./input_ac

Environment Variables

Environment Variables

- A subset of the shell variables are environment 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 by itself will also show all the environment variables.
- The **export** command is used to make a shell variable into an environment variable. E.g. **dog=benji; export dog** creates a new environment variable named dog.
- 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

```

SHELL          LOGNAME      HOME          LANG
SSH_TTY        EUID          PWD
BASH_VERSION   LINES        COLORS        PPID
               IFS
               BASH_ENV    HOSTNAME
MAILCHECK      SHELLOPTS
USER           BASH         PS4           TERM          PIPESTATUS    GROUPS
HISTFILESIZE   OPTIND
               UID          BASH_VERSINFO
BASH_ARGV      PATH         PS1
SHLVL         tmpid        SSH_CONNECTION
               USERNAME    OSTYPE        HISTFILE
BASH_ARGC     USERNAME
HISTSIZ       BASH_LINENO  LESSOPEN
               OPTERR      SSH_CLIENT
HOSTTYPE      LS_COLORS    CVS_RSH
COLUMNS      INPUTRC      BASH_SOURCE   _            MACHTYPE
PROMPT_COMMAND
DIRSTACK      MAIL         SSH_ASKPASS   G_BROKEN_FILENAMES

```

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

Environment Variables

SHELL **SSH_TTY** **LOGNAME** **HOME** **LANG**
 BASH_VERSION EUID PWD
 MAILCHECK consoletype IFS LINES COLORS PPID
USER BASH PS4 **BASH_ENV** **HOSTNAME**
 HISTFILESIZE OPTIND PIPESTATUS GROUPS
 BASH_ARGV **PATH** UID BASH_VERSINFO
SHLVL tmpid **SSH_CONNECTION** PS1
 BASH_ARGC **USERNAME** OSTYPE HISTFILE
HISTSIZ OPTERR BASH_LINENO **LESSOPEN**
 HOSTTYPE **LS_COLORS** **SSH_CLIENT** **CVS_RSH**
 COLUMNS **INPUTRC** BASH_SOURCE MACHTYPE
 PROMPT_COMMAND **MAIL** **SSH_ASKPASS** PS2
 DIRSTACK **G_BROKEN_FILENAMES**

Use *env* to see which of the shell variables have been exported and are also 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 <code>cd</code> 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
```

Shell (Environment) Variables

export command – show all exported variables

```
[rododyduk@opus ~]$ export
declare -x BASH_ENV="/home/cis90/rododyduk/.bashrc"
declare -x CVS_RSH="ssh"
declare -x G_BROKEN_FILENAMES="1"
declare -x HISTSIZE="1000"
declare -x HOME="/home/cis90/rododyduk"
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="rododyduk"
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/rododyduk"
declare -x OLDPWD
declare -x
PATH="/usr/kerberos/bin:/usr/local/bin:/bin:/usr/bin:/home/cis90/rododyduk/./bin:/home/cis90/rododyduk/bin:."
declare -x PWD="/home/cis90/rododyduk"
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="rododyduk"
declare -x USERNAME=""
```

Shell (Environment) Variables

export command – show all exported variables

To create your own environment variable use the export command

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

*There are currently 24
environment variables*

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

```
/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*

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

*Now there are 25
environment variables*

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

*fan is now one of the
environment variables*

Flashback

PS1

bash shell tip

Change PS1 to change the shell prompt

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

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

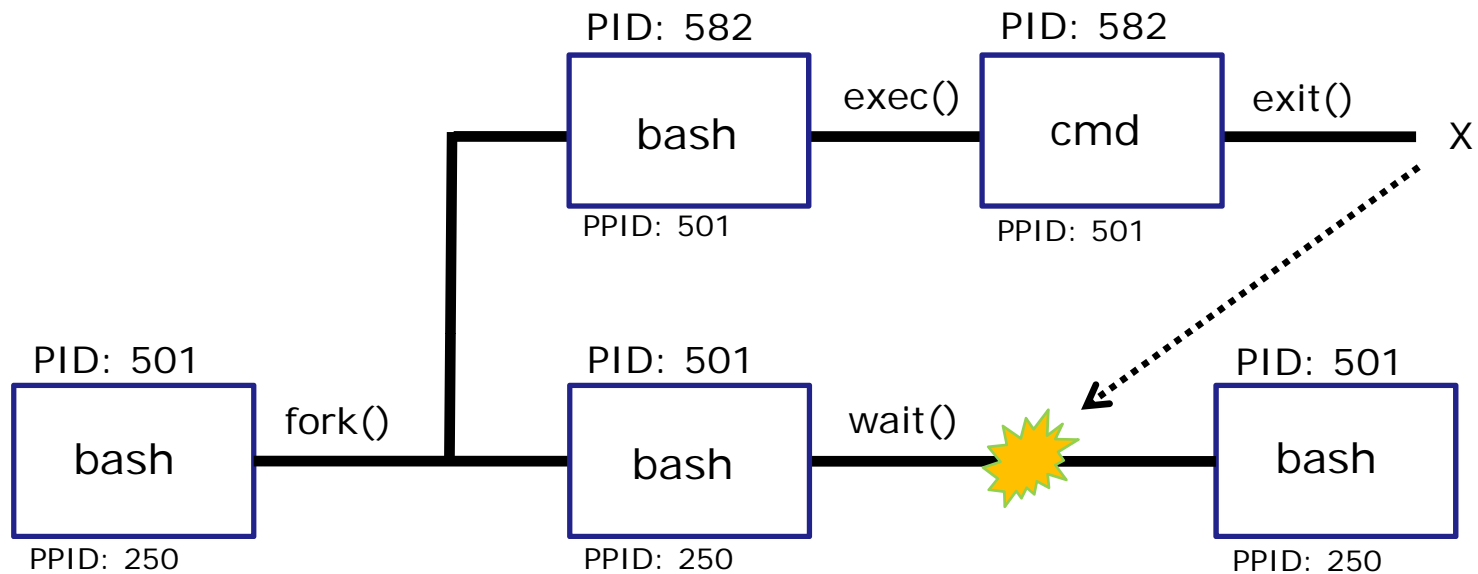
Prompt Code	Meaning
\!	history command number
\#	session command number
\d	date
\h	hostname
\n	new line
\s	shell name
\t	time
\u	user name
\w	entire path of working directory
\W	only working directory
\\$	\$ or # (for root user)

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



Shell Environment

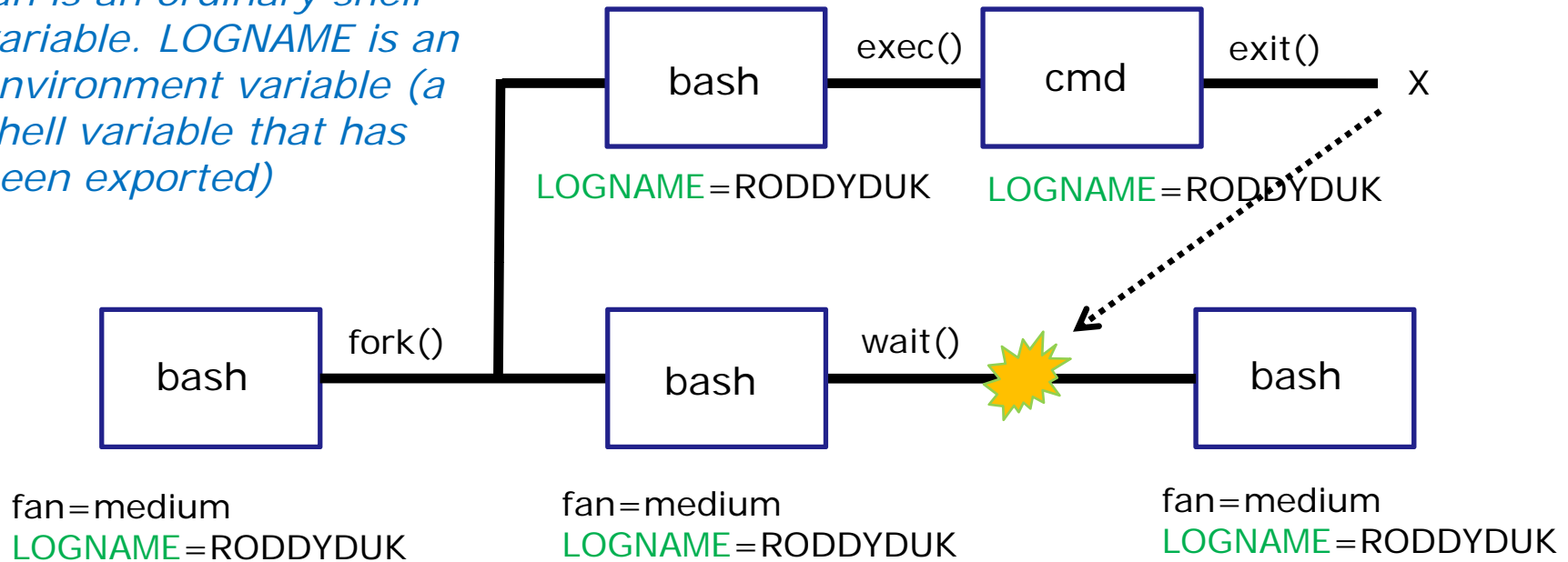
exporting variables



- 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. LOGNAME is an environment variable (a shell variable that has been exported)



- 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.

The child gets the value of LOGNAME only.

Example: Only exported variables are available to the child

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

*fan is a shell variable
LOGNAME is an environment
variable (it has been exported)*

parent

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

*LOGNAME shows up in both
env and set output*

*fan only shows up in set
output*

child

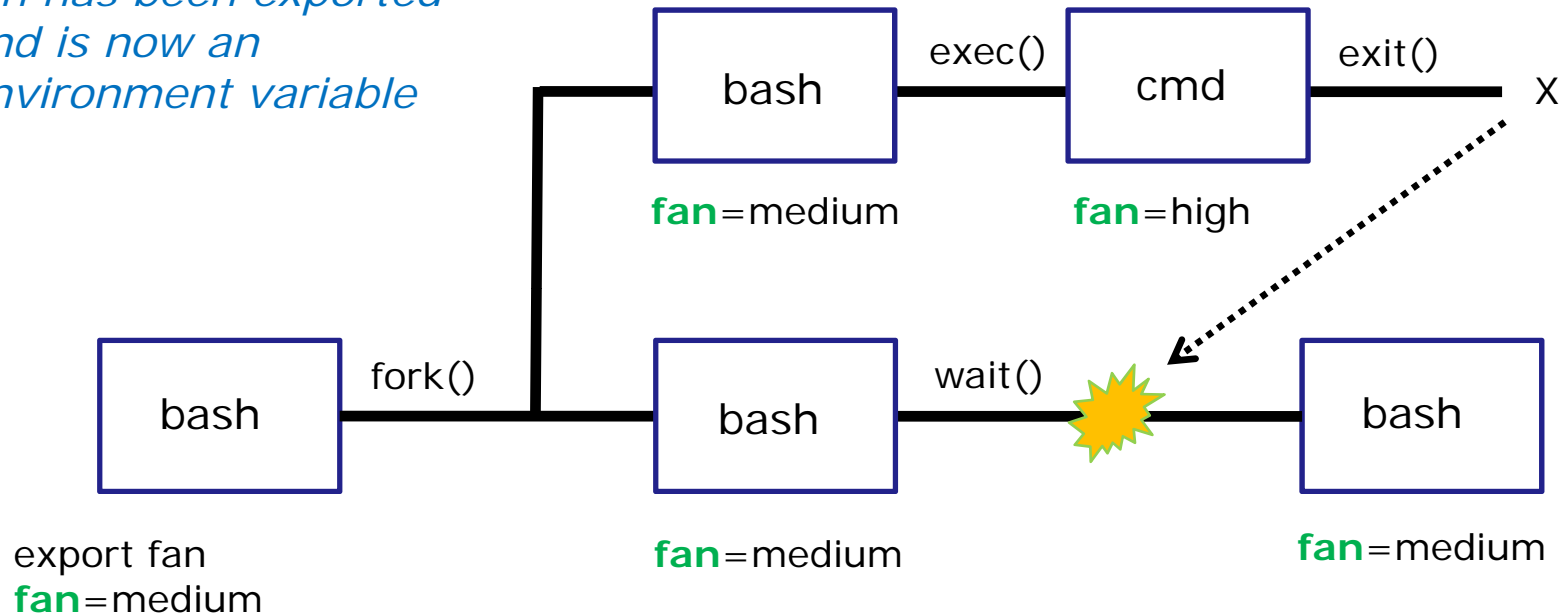
```
/home/cis90/roddyduk $ bash
[roddyduk@opus ~]$ echo $fan $LOGNAME
roddyduk
[roddyduk@opus ~]$ exit
exit
/home/cis90/roddyduk $
```

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

*Only LOGNAME, an environment 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

parent	<pre> /home/cis90/roddyduk \$ echo \$fan medium /home/cis90/roddyduk \$ export fan </pre>	<p><i>export fan so it is available to children</i></p>
child	<pre> /home/cis90/roddyduk \$ bash [roddyduk@opus ~]\$ echo \$fan medium [roddyduk@opus ~]\$ fan=high [roddyduk@opus ~]\$ echo \$fan high [roddyduk@opus ~]\$ exit exit </pre>	<p><i>fan is now available to the child process</i></p> <p><i>the child modifies the variable</i></p>
parent	<pre> /home/cis90/roddyduk \$ echo \$fan medium </pre>	<p><i>any changes made by a child will not effect the parent's variable</i></p>

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`.

alias command

Example 1

We make an alias for the cp command and use it

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

alias command

Example 1 continued

We make an alias for the cp command and use it

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

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

*The **type** command shows it is an alias to the cp command*

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

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

```
/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 2

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

```
/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 $
```

alias command

Example 2 continued

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

```
/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 $
```

```
/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*

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

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

alias an alias

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

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

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

```
Melba Metcher: Hello Patrick!
Here I am at Camp Granada. Things are very interesting,
and they say we'll have some fun when it stops raining.
All the youngsters here are waiting, and the lake has
alligators! You remember "Lambert's Island"? We got
promised swimming last night after dinner.
Now I don't want this to scare you, but my bunk mate has
written: you remember Jeffrey Hardy? They want to
organize a swimming party.
Take me home, oh Melba, Patrick, take me home! I hate Granada.
Now I leave me out in the front where I might get eaten.
I'm a leet. This is dumb. I promise that I won't make noise.
So when the house with other boys, oh please don't make me
stay -- I'm home now and make my.
Dearest Patrick, darling Melba, how a my precious little
loveliness! I will come home to you nice me. I will come
let don't disturb my and kiss me!
Wait a minute! It's stopped raining! They are swimming!
They are swimming! Playing basketball, see that's better!
Melba, Patrick, kindly disregard this letter.
Alan Metcher
```

reduced sized to fit on page

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

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

```
Melba Metcher: Hello Patrick!
Here I am at Camp Granada. Things are very interesting,
and they say we'll have some fun when it stops raining.
All the youngsters here are waiting, and the lake has
alligators! You remember "Lambert's Island"? We got
promised swimming last night after dinner.
Now I don't want this to scare you, but my bunk mate has
written: you remember Jeffrey Hardy? They want to
organize a swimming party.
Take me home, oh Melba, Patrick, take me home! I hate Granada.
Now I leave me out in the front where I might get eaten.
I'm a leet. This is dumb. I promise that I won't make noise.
So when the house with other boys, oh please don't make me
stay -- I'm home now and make my.
Dearest Patrick, darling Melba, how a my precious little
loveliness! I will come home to you nice me. I will come
let don't disturb my and kiss me!
Wait a minute! It's stopped raining! They are swimming!
They are swimming! Playing basketball, see that's better!
Melba, Patrick, kindly disregard this letter.
Alan Metcher
```

reduced sized to fit on page

```
/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

An alias can be made using another alias

single and double quotes

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

double

single

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

```
$ alias p
```

```
alias p='echo on medium off'
```

```
$ p
```

```
on medium off
```

```
$ ac=off
```

```
$ p
```

```
on medium off
```

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

```
$ alias p
```

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

```
$ p
```

```
on medium off
```

```
$ ac=off
```

```
$ p
```

```
off medium off
```

Note: using single quotes prevents bash from expanding the variables when setting 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 (all)

- o adds root's special path

/etc/profile.d/*.sh (all)

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

.bash_profile (user specific)

- o adds user's bin to path

.bashrc (user specific)

- o add aliases here

/etc/bashrc (all)

- o changes umask to 0002 for regular users
- o 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, running the user's `.bashrc` file, running basic commands like `umask` and `set -o` options.

.bash_profile (runs only at login)

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

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

# User specific environment and startup programs

PATH=$PATH:$HOME/../../bin:$HOME/bin:.          Adds the user's bin and
current directories to the path
BASH_ENV=$HOME/.bashrc
USERNAME=""          These variables are set
PS1='$PWD $ '      Prompt (PS1) is defined
export USERNAME BASH_ENV PATH          These variables are exported
umask 002          umask value is set
set -o ignoreeof    EOF's are ignored
stty susp ^F      Suspend character redefined from Z to F
eval `tset -s -m vt100:vt100 -m :\?${TERM:-ansi} -r -Q`  Terminal
type is set

[roddyduk@opus ~]$
```

.bashrc

.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.

```
alias bye="clear; exit"  
alias rm="rm -i"
```

.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    sources the /etc/bashrc file
fi
alias print="echo -e"    creates a print alias, the -e option
                          enables interpretation of backslash
                          escapes
[roddyduk@opus ~]$
```


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 is unable to affect the login shell environment.

Sometimes it is desirable 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.

`.` *myscript* or `source` *myscript*

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 shell will terminate upon the exiting of that command:

`exec clear`

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



Wrap up

Lab 10 - the last one!

The screenshot shows a web browser window with the address bar displaying `simms-teach.com/docs/cis90/cis90lab10.html`. The page content is as follows:

CIS 90 Linux Lab Exercise
CIS 90 Lab 10: The Shell Environment
 Fall 2010

Objectives

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

If you get stuck, have a question or want to share something you learned with this lab then use the CIS 90 Forum at <http://opus.cabrillo.edu/forum/viewforum.php?f=25>

Procedure

Log on to Opus and start this lab from your home directory.


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 `envx` command to see if it is in your environment. Is it there? What must you do to put it in the

The browser's download bar at the bottom shows a file named `cis-90-TEST-2-Fall-1....pdf` and a button to `Show all downloads...`

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 2 points extra credit, email risimms@cabrillo.edu your answers by noon tomorrow (November 18, 2010)

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
```

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

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

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

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

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

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

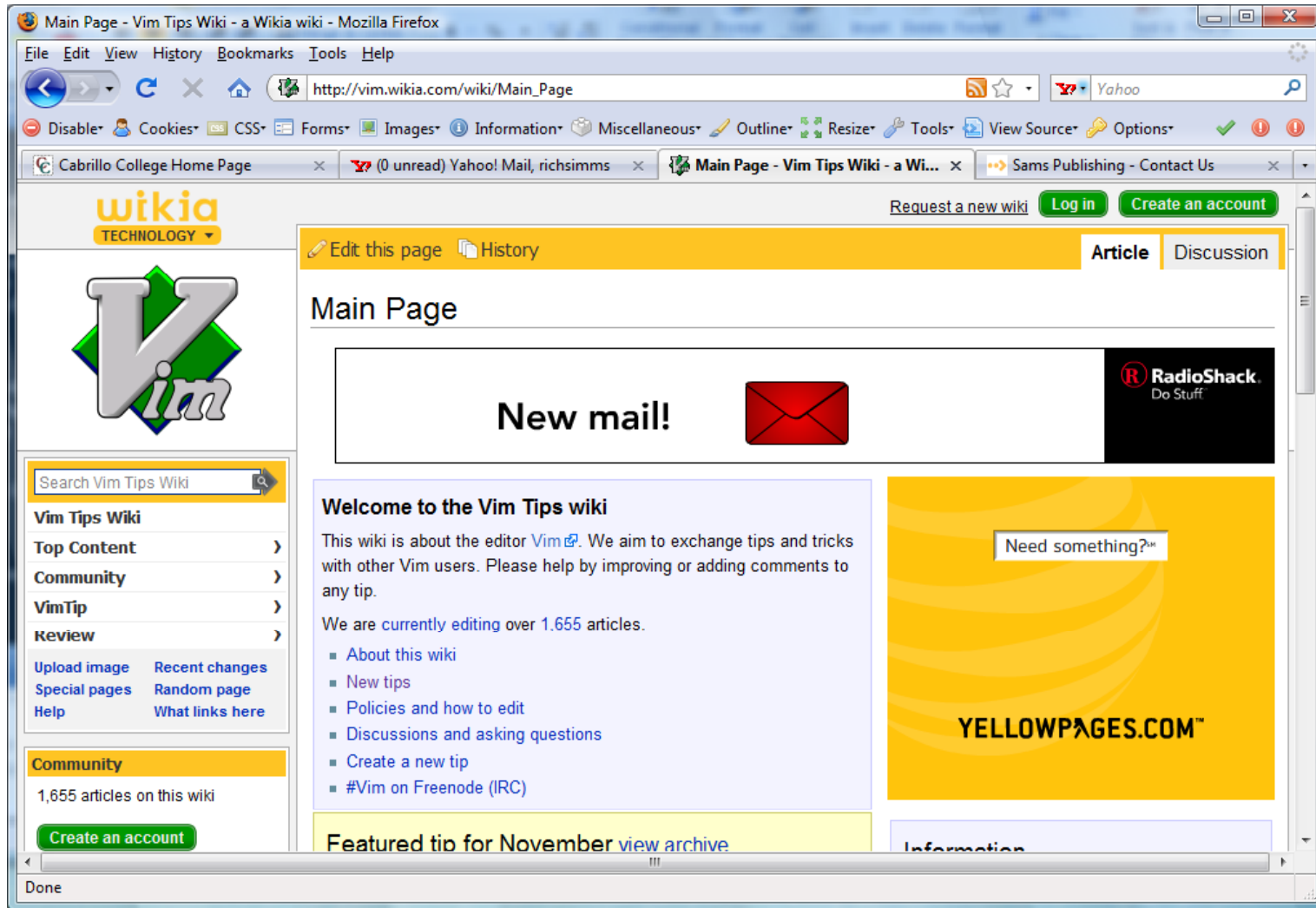
display the help message

-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

File Edit View History Bookmarks Tools Help

http://nostarch.com/mug.htm


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



The Mug of Vi
12 ounces
heavy-duty

\$12.95

Order now

Hydration
harmony

Copyright

See mug text

Done

Big Mug Label - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://nostarch.com/mug_big.htm

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

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

FILE COMMANDS	DELETING /INSERTING TEXT	
vi filename(s) edit a file or files	de, dd, x delete word, line, character	0 go to beginning of line (zero)
vi -r filename retrieve saved file after crash	ndd, nxx delete <i>n</i> lines, <i>n</i> characters), (move to next, previous sentence
ZZ, :wq, :x save and exit	x, X delete character forward, backward	}, { move forward, back one word
:q, :q! quit; quit without saving	D, d\$ delete to end of line	w, b go to end of current or next word
:w, :w filename save file, save file as <i>fn</i>	dmotion delete from cursor to <i>motion</i> (<i>\$</i> , <i>0</i> , etc.)	e go to end of current or next word
:e filename edit <i>filename</i>		CUT / COPY / PASTE
:r filename insert <i>filename</i>		yy, nY copy <i>n</i> lines
:sh drop to shell		yw, yy copy word, line
:!cmd run command <i>cmd</i>		p, P paste text after, before cursor
:r !cmd execute <i>cmd</i> and insert output		a, i insert text after, before cursor
/txt, /txt find <i>txt</i> forward or backward		A, I insert text end, beginning of line
/*txt find next line that starts with <i>txt</i>		WICKED COOL STUFF
ns, N repeat last search backward, forward		~ change case
R replace text from current character		xp transpose characters
		J combine current line with next
		mp create a mark called <i>p</i>
		~ return to <i>p</i>
		d'x, y'x delete, copy text from mark to cursor
		:> n indent <i>n</i> lines

Done