



Lesson Module Checklist

- Slides
- WB converted

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

- LabX1 and Project posted
- Timer lock set on turnin directory

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

Introductions and Credits



Jim Griffin

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



Rich Simms

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

And thanks to:

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



Instructor: **Rich Simms**

Dial-in: **888-886-3951**

Passcode: **136690**



Francisco



Chris



Justin



Jesus



Shenghong



Paul



Roberto



Sam



Navin



Jimmy



Luis



Tommy



Abraham



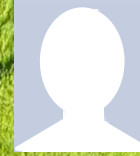
Ann



Cameron



Cody



Alejandrino



Deane



Nadia



Richard Z.



Gabriel



Ryan



Takashi



Jeff



Nick



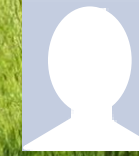
Jonathan



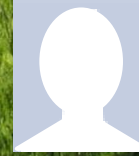
Shea



Matthew



James



Richard I.



Nicole

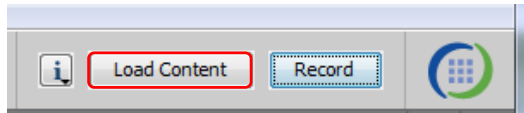


Aaron



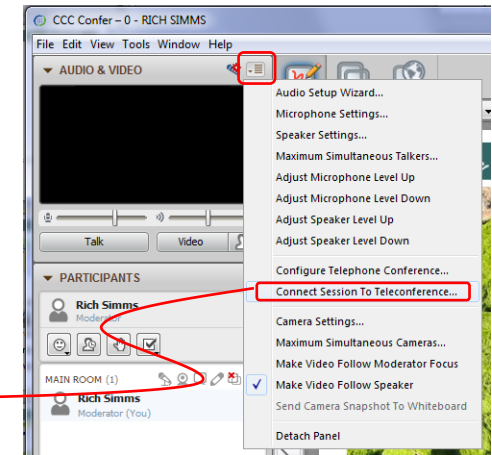
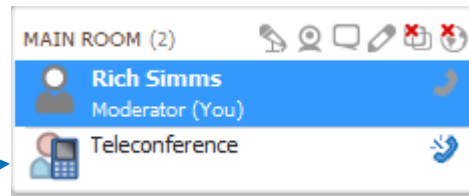
Instructor CCC Confer checklist

[] Preload White Board



[] Connect session to Teleconference

Session now connected to teleconference



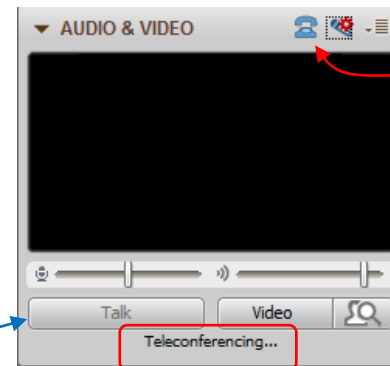
[] Is recording on?



Red dot means recording

[] Use teleconferencing, not mic

Should be greyed out



Should show as this live "off hook" telephone handset icon and the Teleconferencing ... message displayed



Instructor CCC Confer checklist

The screenshot displays a Windows desktop with several applications open:

- CCC Confer**: A video conferencing window on the left showing a participant named Rich Simms. It includes controls for audio and video, a list of participants, and a chat window.
- Chrome**: A browser window in the top right displaying a PDF document titled 'simms-teach.com/docs/cis90/cis-90-TEST-1-Fall-12.pdf'. The document contains flashcard questions and answers. A red box labeled 'chrome' points to the browser window.
- Putty**: A terminal window in the center showing a login session for 'simben90@oslab:~'. The terminal output includes the password prompt, 'Access denied', and a directory listing for the current directory. A red box labeled 'putty' points to the terminal window.
- vSphere Client**: A window in the bottom right showing the vCenter interface for 'CIS VLab'. It displays a tree view of virtual machines and a 'Recent Tasks' table. A red box labeled 'vSphere Client' points to the interface.
- foxit for slides**: A red box labeled 'foxit for slides' points to a PDF viewer window in the background.

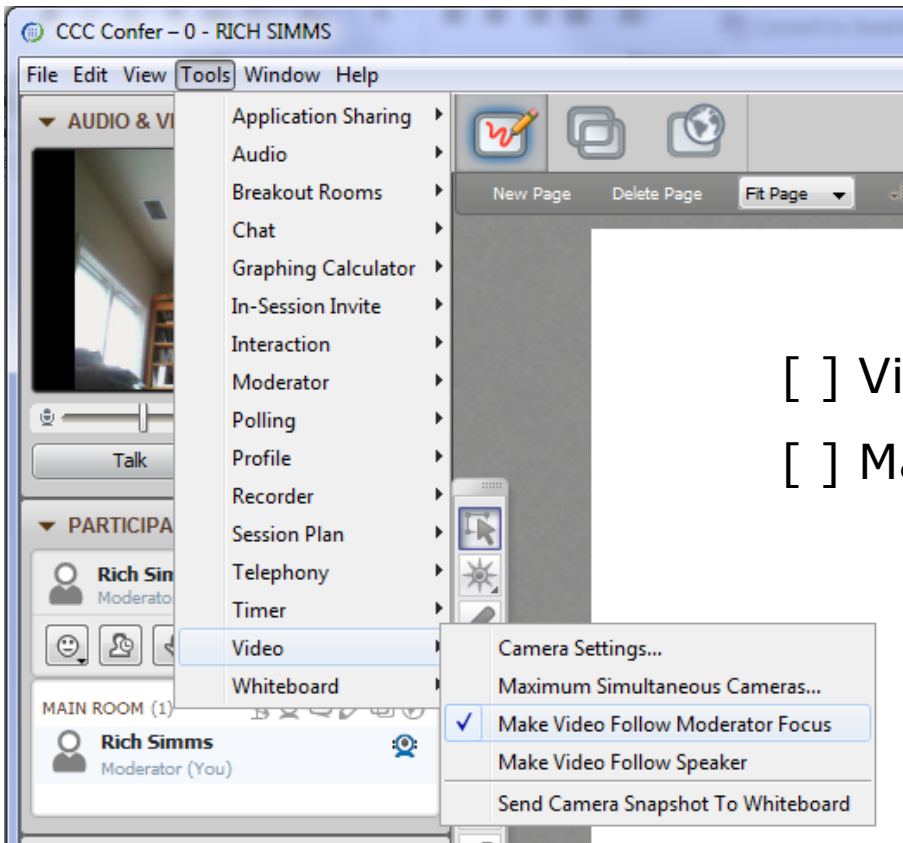
At the bottom of the desktop, the taskbar shows various application icons, and the system tray displays the time as 6:52 AM on 10/10/2012.

[] layout and share apps





Instructor CCC Confer checklist

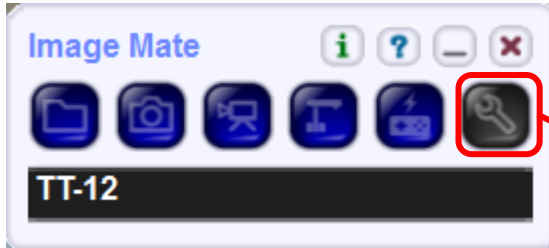


[] Video (webcam)

[] Make Video Follow Moderator Focus



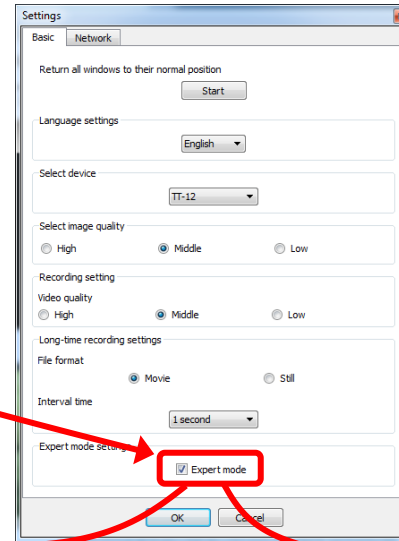
Using Elmo with CCC Confer



Elmo rotated down to view side table



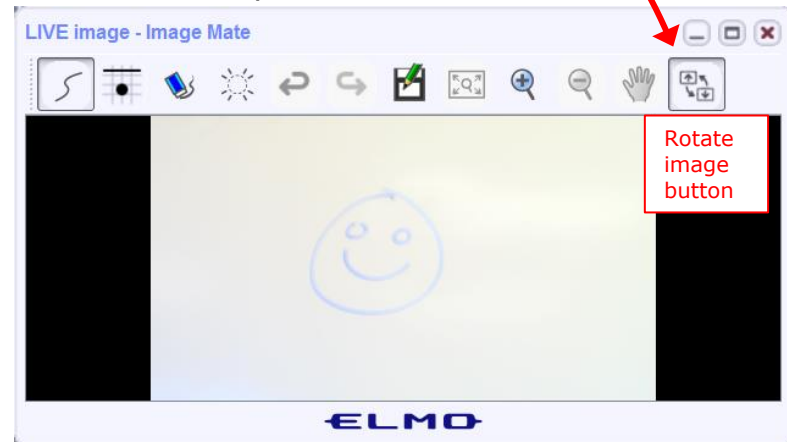
Run and share the Image Mate program just as you would any other app with CCC Confer



The "rotate image" button is necessary if you use both the side table and the white board.

Quite interesting that they consider you to be an "expert" in order to use this button!

Elmo rotated up to view white board



Instructor CCC Confer checklist

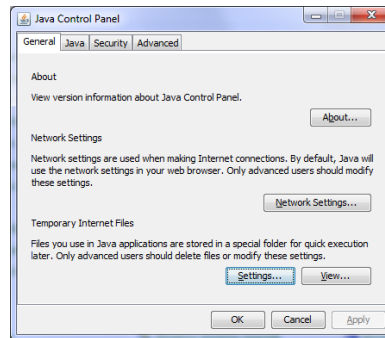
Universal Fix for CCC Confer:

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

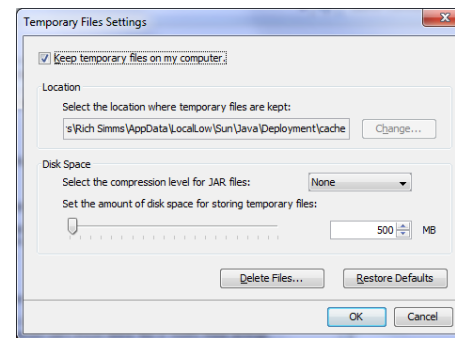
Control Panel (small icons)



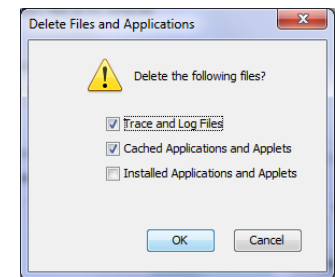
General Tab > Settings...



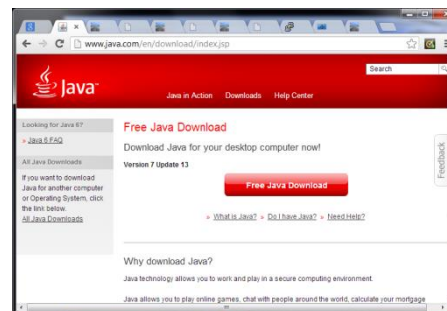
500MB cache size



Delete these



Google Java download



Quiz

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

See electronic white board

email answers to: risimms@cabrillo.edu

(answers must be emailed within the first few minutes of class for credit) 9



Shell Scripting and Printing

Objectives

- Be able to print, view the print queue and cancel print jobs

Agenda

- Quiz
- Housekeeping
- Refresh
- Shell scripting
- Printing



Questions

Questions?

Lesson material?

Labs? Tests?

How this course works?

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

Who questions much, shall learn much, and retain much.

- Francis Bacon

If you don't ask, you don't get.

- Mahatma Gandhi

Chinese
Proverb

他問一個問題，五分鐘是個傻子，他不問一個問題仍然是一個傻瓜永遠。

He who asks a question is a fool for five minutes; he who does not ask a question remains a fool forever.



Breaking things in Lab 10

The path (PATH) variable ... a Review

- Lab 10 often results in clobbered paths and students may think all the commands have disappeared!
- The path is a list of directories each containing commands, programs and scripts.
- The path is used by the shell to locate commands to run.
- The PATH variable defines the directories (separated by ":"s) and the search order.
- If your path gets clobbered it is possible to run commands. However to do that you must specify the full absolute pathname. For example you can always run the **ttty** command as follows:

```
/home/cis90/simben $ /usr/bin/tty  
/dev/pts/0
```

The path (PATH) variable ... a Review

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

```
/usr/lib/qt-3.3/bin:/usr/local/bin:/bin:/usr/bin:/usr/local/sbin:
```

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

1. What is the fourth directory on this path?
2. Can you name the first command, in alphabetic order, found in this directory?

Put your answers in the chat window

Backup and remove your path

```
/home/cis90/simben $ oldpath=$PATH  
/home/cis90/simben $ unset PATH
```

Backup your current path

```
/home/cis90/simben $ tty  
-bash: tty: No such file or directory
```

The tty command can no longer be run by typing just its name

```
/home/cis90/simben $ /usr/bin/tty  
/dev/pts/0
```

Instead the full absolute pathname must be used

```
/home/cis90/simben $ PATH=$oldpath  
/home/cis90/simben $ tty  
/dev/pts/0
```

Restore your path to what it was

Class Activity

Backup and remove your path variable:

```
/home/cis90/simben $ oldpath=$PATH
```

```
/home/cis90/simben $ unset PATH
```

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

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

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

Don't restore your path yet. We will build it up one directory at a time

Making a path

```
/home/cis90/simben $ ls letter
```

```
-bash: ls: No such file or directory
```



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

```
letter
```

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

*A temporary workaround
is to specify the full path
to the command*

Making a path

Some commands still work without a path ... why?

```
/home/cis90/simben $ echo "I want my path back"  
I want my path back
```

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

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

Making a path

Fixing the path, one directory at a time ...

```
/home/cis90/simben $ ls letter
-bash: ls: No such file or directory
```



/home/cis90/simben \$ **PATH=/bin** *The ls command is in /bin so lets put that on the path*

```
/home/cis90/simben $ ls letter
letter
```



```
/home/cis90/simben $ stat letter
-bash: stat: command not found
```



/home/cis90/simben \$ **PATH=\$PATH:/usr/bin**

```
/home/cis90/simben $ stat letter
```

The stat command is in /usr/bin so lets append that directory too

```
File: `letter'
Size: 1059          Blocks: 16          IO Block: 4096
regular file
Device: fd00h/64768d    Inode: 102594      Links: 1
Access: (0644/-rw-r--r--)  Uid: ( 1000/simben90)  Gid: (
90/  cis90)
Access: 2012-04-30 15:43:28.000000000 -0700
Modify: 2012-03-20 10:31:30.000000000 -0700
Change: 2012-04-30 07:34:30.000000000 -0700
```

Making a path

```
/home/cis90/simben $ allscripts  
-bash: allscripts: command not found
```



*The **allscripts** shell script is in /home/cis90/bin so let's add that directory to the path as well*



```
/home/cis90/simben $ PATH=$PATH:/home/cis90/bin  
/home/cis90/simben $ allscripts
```

```
*****  
*                               Fall 2012 CIS 90 Online Projects                               *  
*****  
1) Andrew  
2) Ben  
3) Benji  
4) Bryn  
5) Carlile  
6) Carlos  
  <snipped>  
21) Ray  
22) Rita  
23) Sean C.  
24) Sean F.  
25) Shahram  
  
99) Exit  
  
Enter Your Choice:
```

Making a path

```
/home/cis90/simben $ datecal  
bash: datecal: command not found
```



The **datecal** shell script is in your own bin directory so lets add that to the path as well



```
/home/cis90/simben $ PATH=$PATH:/home/cis90/simben/bin  
/home/cis90/simben $ datecal
```

```
Tue May 8 14:30:59 PDT 2012
```

```

      April 2012
Su Mo Tu We Th Fr Sa
  1  2  3  4  5  6  7
  8  9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30

```

```

      May 2012
Su Mo Tu We Th Fr Sa
      1  2  3  4  5
  6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31

```

```

      June 2012
Su Mo Tu We Th Fr Sa
      1  2
  3  4  5  6  7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30

```

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

Making a path

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

```
-bash: dogbone: command not found
```



```
/home/cis90/simben $ ./dogbone
```

```
What is your name? Benji
```

```
What is your favorite bone? Chicken
```

```
Hi Benji, your favorite bone is Chicken
```

A temporary workaround is to put a ./ in front of the command

How can I run a script in the current directory without having to put a ./ in front of it?

Making a path

Easy ... add the "." directory to the path

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

```
-bash: dogbone: command not found
```



```
/home/cis90/simben $ PATH=$PATH:.
```

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

```
What is your name? Benji
```

```
What is your favorite bone? Chicken
```

```
Hi Benji, your favorite bone is Chicken
```



Making a path

Rebuilding the path by appending directories one at a time

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

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

Start with /bin which has all the essential UNIX/Linux commands

```
/home/cis90/simben $ PATH=$PATH:/usr/bin
/home/cis90/simben $ echo $PATH
/bin:/usr/bin
```

Append /usr/bin which has hundreds of additional UNIX/Linux commands

```
/home/cis90/simben $ PATH=$PATH:/home/cis90/bin
/home/cis90/simben $ echo $PATH
/bin:/usr/bin:/home/cis90/bin
```

Append the CIS 90 class bin directory

```
/home/cis90/simben $ PATH=$PATH:/home/cis90/simben/bin
/home/cis90/simben $ echo $PATH
/bin:/usr/bin:/home/cis90/bin:/home/cis90/simben/bin
```

Append your own student bin directory

```
/home/cis90/simben $ PATH=$PATH:.
/home/cis90/simben $ echo $PATH
/bin:/usr/bin:/home/cis90/bin:/home/cis90/simben/bin:.
```

Append the current directory

└──────────────────┘
└──────────────────┘
└──┘
CIS 90 class bin directory

 Student bin directory

 Current directory

.bash_profile

Making the path permanent using .bash_profile

```

/home/cis90/simben $ cat .bash_profile
# .bash_profile

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

# User specific environment and startup programs

PATH=$PATH:/home/cis90/bin:$HOME/bin:.
BASH_ENV=$HOME/.bashrc
USERNAME=""
PS1='$PWD $ '
export USERNAME BASH_ENV PATH
umask 002
set -o ignoreeof
stty susp
eval `tset -s -m vt100:vt100 -m :\?${TERM:-ansi} -r -Q `

/home/cis90/simben $

```

This customizes the normal path by appending the class bin directory, the student's bin directory and the "current" directory



Extra Credit Special Answer



Extra Credit Special (from Lesson 12)

1) *Why did the prompt change?*

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

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 answers to **both** questions before the Lesson 13 class starts



Lesson 12

Review

The rules of the road for variables

Process Rule #1: When a shell forks a child, only copies of exported variables are made available to the child.

Process Rule #2: A child can modify the variables it receives but those modifications will not change the parent's variables.

▪ and **SOURCE**

`▪ <script-name>`
`source <script-name>` } *equivalent*

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, and then continue on where it left off.

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

exec command

exec *<command>*

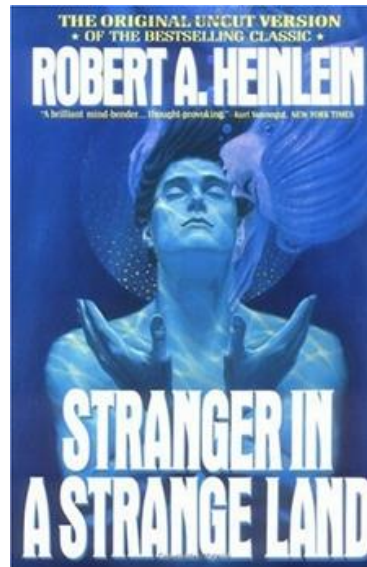
If a UNIX command is run using the **exec** *<command>*, the bash code in the process is overlaid by the *<command>* code, when finished the process will terminate.

For example:

exec clear

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

grok that?



The flowers script /home/cis90/bin/flowers

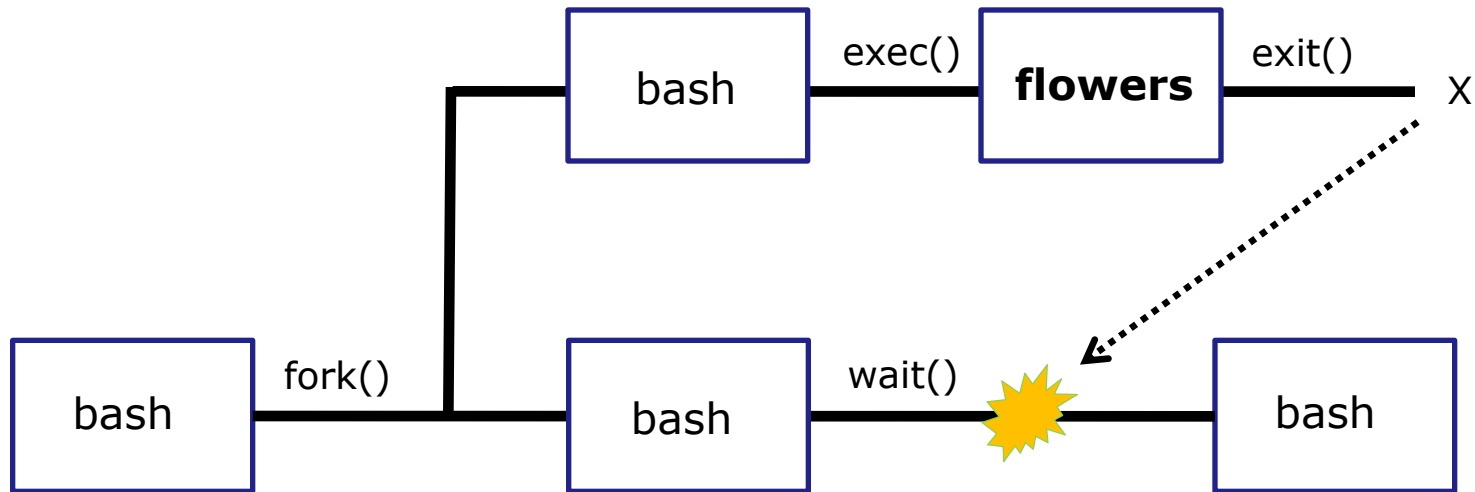
```
#!/bin/bash
#
# Useful alias:
#   alias go='echo roses are \"$roses\" and violets are \"$violets\"'
#
echo
echo "==> Entering child process <=="
ps -f
echo "==> showing variables in child <=="
echo "  " roses are '$roses'
echo "  " violets are '$violets'
echo "==> setting variables in child <=="
roses=black
violets=orange
echo "  " roses are '$roses'
echo "  " violets are '$violets'
echo "==> Leaving child process <=="
echo
```

Show the parent, child and the ps processes

Show the values of the roses and violets variables

Set the values of the roses and violets variables to new values

The flowers script /home/cis90/bin/flowers



Use the **flowers** script to test your understanding of how variables are handled with child processes

The flowers script /home/cis90/bin/flowers

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

```
==> Entering child process <==
```

UID	PID	PPID	C	STIME	TTY	TIME	CMD
simben90	17518	17512	0	08:32	pts/0	00:00:00	-bash
simben90	17568	17518	0	08:33	pts/0	00:00:00	/bin/bash /home/cis90/bin/flowers
simben90	17575	17568	8	08:33	pts/0	00:00:00	ps -f

```
==> showing variables in child <==
```

```
roses are ""
```

```
violets are ""
```

```
==> setting variables in child <==
```

```
roses are "black"
```

```
violets are "orange"
```

```
==> Leaving child process <==
```

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

```
#!/bin/bash
#
# Useful alias:
# alias go='echo roses are \"$roses\" and violets are \"$violets\"'
#
echo
echo "==> Entering child process <=="
ps -f
echo "==> showing variables in child <=="
echo " " roses are '$roses'
echo " " violets are '$violets'
echo "==> setting variables in child <=="
roses=black
violets=orange
echo " " roses are '$roses'
echo " " violets are '$violets'
echo "==> Leaving child process <=="
echo
```

Create an alias to show variable values

Note, the double quotes are escaped. We don't want bash to treat them as special metacharacters. We just want the double quotes preserved so they can be seen in the output of the echo command.

```
/home/cis90/simben $ alias go='echo roses are \"$roses\" and violets are \"$violets\"'
```

```
/home/cis90/simben $ alias go  
alias go='echo roses are \"$roses\" and violets are \"$violets\"'
```

```
/home/cis90/simben $ go  
roses are "" and violets are ""
```

Since there are no shell variables named roses or violets the echo command prints nothing for them.

Use the alias to show the values of the two variables

```
/home/cis90/simben $ go  
roses are "" and violets are ""
```

```
/home/cis90/simben $ roses=red  
/home/cis90/simben $ go  
roses are "red" and violets are ""
```

*Now the roses variable
has been created and
initialized*

```
/home/cis90/simben $ violets=blue  
/home/cis90/simben $ go  
roses are "red" and violets are "blue"
```

*Now the violets variable
has been created and
initialized*

Use the alias to show the values of the two variables

```
/home/cis90/simben $ unset roses  
/home/cis90/simben $ go  
roses are "" and violets are "blue"
```

*Now the roses
variable no longer
exists*

```
/home/cis90/simben $ unset violets  
/home/cis90/simben $ go  
roses are "" and violets are ""
```

*Now the violets
variable no longer
exists*

Activity

```
/home/cis90/simben $ roses=red; violets=blue  
/home/cis90/simben $ go  
roses are "red" and violets are "blue"  
/home/cis90/simben $ env | grep roses  
/home/cis90/simben $ env | grep violets  
/home/cis90/simben $ flowers
```

Will the flowers script see the values of the roses and violets variables?

Write your answer in the chat window

***NO**, the roses and violets variables were not exported*

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

```
==> Entering child process <==
```

UID	PID	PPID	C	STIME	TTY	TIME	CMD
simben90	25106	25059	0	17:16	pts/8	00:00:00	-bash
simben90	27052	25106	0	17:19	pts/8	00:00:00	/bin/bash /home/cis90/bin/flowers
simben90	27059	27052	0	17:19	pts/8	00:00:00	ps -f

```
==> showing variables in child <==
```

```
roses are ""  
violets are ""
```

The child cannot view the values of the parent's non-exported variables (Rule #1)

```
==> setting variables in child <==
```

```
roses are "black"  
violets are "orange"
```

```
==> Leaving child process <==
```

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

Activity

```
/home/cis90/simben $ roses=red; violets=blue
/home/cis90/simben $ export roses
/home/cis90/simben $ env | grep roses
roses=red
/home/cis90/simben $ env | grep violets
/home/cis90/simben $ go
roses are "red" and violets are "blue"
/home/cis90/simben $ flowers
```

Will the flowers script see the value of the roses variable or the violets variable?

Write your answer in the chat window



Yes, the flowers script can see the roses variable now which was exported

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

```
==> Entering child process <==
```

UID	PID	PPID	C	STIME	TTY	TIME	CMD
simben90	25106	25059	0	17:16	pts/8	00:00:00	-bash
simben90	32147	25106	0	17:27	pts/8	00:00:00	/bin/bash /home/cis90/bin/flowers
simben90	32154	32147	0	17:27	pts/8	00:00:00	ps -f

```
==> showing variables in child <==
```

```
roses are "red"
```

```
violets are ""
```

The child now sees the value of roses but not violets (Rule #1)

```
==> setting variables in child <==
```

```
roses are "black"
```

```
violets are "orange"
```

```
==> Leaving child process <==
```

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

Activity

```
/home/cis90/simben $ roses=red; violets=blue
/home/cis90/simben $ export roses violets
/home/cis90/simben $ env | grep roses
roses=red
/home/cis90/simben $ env | grep violets
violets=blue
/home/cis90/simben $ go
roses are "red" and violets are "blue"
/home/cis90/simben $ flowers
```

Will the flowers script change the values of the roses and violets variables?

Write your answer in the chat window



No, the flowers script which runs as a child process cannot change the parent's variables

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

```
==> Entering child process <==
```

UID	PID	PPID	C	STIME	TTY	TIME	CMD
simben90	28732	28724	0	17:51	pts/0	00:00:00	-bash
simben90	29383	28732	0	18:11	pts/0	00:00:00	/bin/bash /home/cis90/bin/flowers
simben90	29390	29383	0	18:11	pts/0	00:00:00	ps -f

```
==> showing variables in child <==
```

```
roses are "red"
```

```
violets are "blue"
```

```
==> setting variables in child <==
```

```
roses are "black"
```

```
violets are "orange"
```

The child can only change copies of the parents variables

```
==> Leaving child process <==
```

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

```
roses are "red" and violets are "blue"
```

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

The child cannot change the parent's variables (Rule #2)

Activity

```
/home/cis90/simben $ roses=red; violets=blue
/home/cis90/simben $ export roses violets
/home/cis90/simben $ env | grep roses
roses=red
/home/cis90/simben $ env | grep violets
violets=blue
/home/cis90/simben $ go
roses are "red" and violets are "blue"
/home/cis90/simben $ . flowers
```

Will the flowers script change the values of the roses and violets variables?

Write your answer in the chat window

Yes, if sourced, flowers will not run as a child process and can change the parent's variables

```
/home/cis90/simben $ . flowers
```

```
==> Entering child process <==
```

UID	PID	PPID	C	STIME	TTY	TIME	CMD
simben90	28732	28724	0	17:51	pts/0	00:00:00	-bash
simben90	29480	28732	0	18:15	pts/0	00:00:00	ps -f

```
==> showing variables in child <==
```

```
    roses are "red"
```

```
    violets are "blue"
```

```
==> setting variables in child <==
```

```
    roses are "black"
```

```
    violets are "orange"
```

```
==> Leaving child process <==
```

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

```
roses are "black" and violets are "orange"
```

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

```

/home/cis90/rodduk $ 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
eval `tset -s -m vt100:vt100 -m`

/home/cis90/rodduk $
    
```

And now you know why the bash login scripts are sourced rather than run as child processes.

*Note: the . (dot) and **source** commands are equivalent*

```

/home/cis90/rodduk $ cat .bashrc
# .bashrc

# User specific aliases and functions

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


Activity

```
/home/cis90/simben $ roses=red; violets=blue
/home/cis90/simben $ export roses violets
/home/cis90/simben $ env | grep roses
roses=red
/home/cis90/simben $ env | grep violets
violets=blue
/home/cis90/simben $ go
roses are "red" and violets are "blue"
/home/cis90/simben $ exec flowers
```

What will happen if flowers is exec'ed?

Write your answer in the chat window



The flowers script run to completion and
your session ends



Housekeeping

Previous material and assignment

1. Lab 10 due by 11:59PM tonight
2. Use the **check10** script to check your work
3. After you submit your lab10 file you may comment out your riddle command in `.bash_profile`
4. The Extra Credit Labs X1 and X2 (30 points each) are available.
5. The Final Project is available and due in **two weeks**.

Spring 2015 Linux Classes and Prerequisites

CIS 90 Introduction to UNIX/Linux

Provides a technical overview of the UNIX/Linux operating system, including hands-on experience with commands, files, and tools. Prerequisite: CIS 72. Transfer Credit: CSU.

Section	Days	Times	Units	Instructor	Room
88445	W	09:00AM-12:05PM	3.00	R.Simms	OL
&	Arr.	Arr.		R.Simms	OL

Section 88445 is an ONLINE course. Meets weekly throughout the semester online during the scheduled times by remote technology with an additional 50 min arranged online lab per week. For details, see instructor's web page at go.cabrillo.edu/online.

Section	Days	Times	Units	Instructor	Room
88446	W	09:00AM-12:05PM	3.00	R.Simms	828
&	Arr.	Arr.		R.Simms	OL

Section 88446 is a Hybrid ONLINE course. Meets weekly throughout the semester at the scheduled times with an additional 50 min online lab per week. For details, see instructor's web page at go.cabrillo.edu/online.

Transfer Credit: CSU.

Section	Days	Times	Units	Instructor	Room
87038	M	05:30PM-09:35PM	4.00	R.Graziani	828
&	Arr.	Arr.		R.Graziani	OL

Section 87038 is a Hybrid ONLINE course. Meets weekly throughout the semester at the scheduled times with an additional 50 min online lab per week. Students will be required to show that they meet the course prerequisites. For details, see instructor's web page at go.cabrillo.edu/online.

CIS 192AB UNIX/Linux Network Administration

Teaches the building of network infrastructures, and the installation, configuration, and protection services on Linux TCP/IP networks. Prerequisites: CIS 81 and CIS 90 or equivalent skills. Recommended Preparation: CIS 191AB

Section	Days	Times	Units	Instructor	Room
88451	Arr.	Arr.	4.00	M.Matera	OL
Section 88451 is an ONLINE course. For details, see instructor's web page at go.cabrillo.edu/online .					
88453	TH	05:30PM-09:35PM	4.00	M.Matera	828
&	Arr.	Arr.		M.Matera	OL

Section 88453 is an Hybrid ONLINE course. Meets weekly throughout the semester at the scheduled times with an additional 50 min online lab per week. Students will be required to show that they meet the course prerequisites. For details, see instructor's web page at go.cabrillo.edu/online.

CIS 193AB UNIX/Linux Security Administration

Teaches how to perform the tasks and examine the strategies of UNIX/Linux host, files, and network security management. Prerequisites: CIS 192AB Recommended Preparation: CIS 175

Section	Days	Times	Units	Instructor	Room
88454	Arr.	Arr.	4.00	M.Matera	OL
Section 88454 is an ONLINE course. For details, see instructor's web page at go.cabrillo.edu/online .					
88455	TH	10:00AM-02:05PM	4.00	M.Matera	829
&	Arr.	Arr.		M.Matera	OL

Section 88455 is a Hybrid ONLINE course. Meets weekly throughout the semester at the scheduled times with an additional 50 min online lab per week. Students will be required to show that they meet the course prerequisites. For details, see instructor's web page at go.cabrillo.edu/online.

Final Exam

Test #3 (final exam) is **THURSDAY** December 18 1:00-3:50PM

	12/18	<p>Test #3 (the final exam)</p> <p>Time</p> <ul style="list-style-type: none"> • 1:00PM - 3:50PM in Room 828 <p>Materials</p> <ul style="list-style-type: none"> • Test (blackboard) <p>CCC Confer</p> <ul style="list-style-type: none"> • Enter virtual classroom • Class archives 		<p>5 posts</p> <p>Lab X1</p> <p>Lab X2</p>
--	-------	--	--	--

- All students will take the test at the same time.
- Working students will need to plan ahead to take time off from work for the test.

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

GRADES

- Check your progress on the Grades page
- If you haven't already, send me a student survey to get your LOR secret code name
- Graded labs & tests are placed in your home directories on Opus
- Answers to labs, tests and quizzes are in the `/home/cis90/answers` directory on Opus

Current Point Tally

Points that could have been earned:

9 quizzes:	27 points
9 labs:	270 points
2 tests:	60 points
3 forum quarters:	60 points
Total:	417 points

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



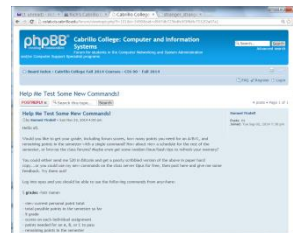
Use Sam's new Python command to see how many points you have earned and how many more you need for the grade you want:

grades <LOR codename>



Use Sam's new Python command to see when the next assignments are due:

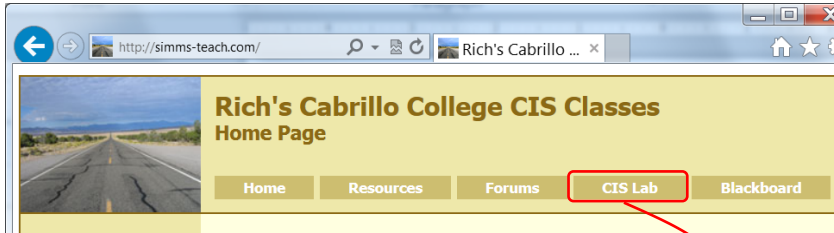
schedule



For more details on these and other commands see Sam's post on the forum

CIS Lab Schedule

<http://webhawks.org/~cislab/>

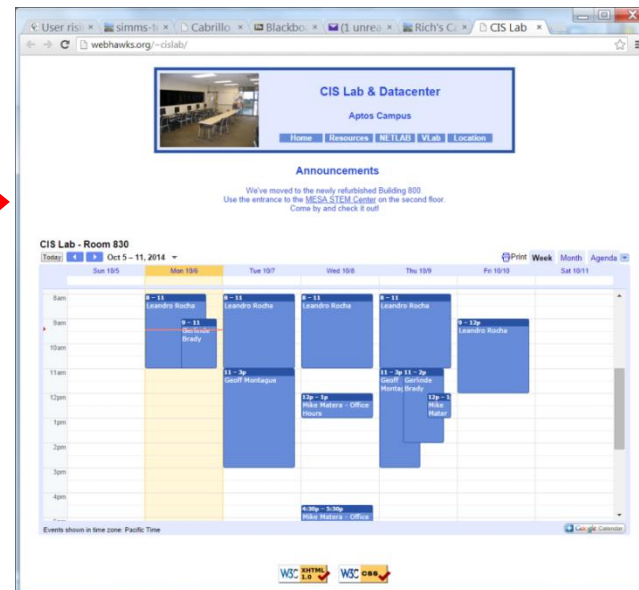


Not submitting tests or lab work?

If you would like some additional help come over to the CIS Lab.

*Leandro, Geoff and Nick
(Mondays 10-4) are all CIS 90
Alumni.*

*Michael is the other Linux
instructor.*



Or hang around after class. Rich has his office hours right after each class in Room 828.

Free CIS 90 Tutoring Available

<http://www.cabrillo.edu/services/tutorials/>

The screenshot shows the website's layout. At the top is the Cabrillo College logo and navigation links. Below is a main content area with several sections:

- TUTORIALS**: Includes an image of students and a list of 'ANNOUNCEMENTS & DEADLINES' for Spring 2014, such as American Sign Language, CABT, CIS, and History 17A.
- Welcome to the Tutorials Center!**: A section describing free peer tutoring services, including appointment-based sessions, weekly sessions, and small group tutoring.
- The following classes are being tutored for Spring 2014:**: A list of classes including Accounting, ASL, Biology, CABT, **Computer and Information Systems (CIS) 81, 90, 172** (highlighted with a red box), and Chemistry.
- CONTACT INFORMATION**: Provides details for the Tutorials Center, including location (Room 1080A), phone (831.479.6470), email (tutorialscenter@cabrillo.edu), and hours (Monday-Thursday: 9am-5pm, Friday: 9am-1pm).



Matt Smithey

All students interested in tutoring in CIS 90, 172, and 81 classes need to come directly to the Tutorials Center to schedule, register and fill out some paperwork. This is just a one-time visit.

The tutoring will take place at the STEM center and they will log in and log out on a computer you have designated (I will figure out exactly what that means).

Matt is available M: 9:00-5:00, T: 9-11 and 2-5, Wed: 9-12 and Th: 9-11 and 3-5.

More Free CIS 90 Tutoring Available

The screenshot shows a web browser window displaying a forum post on the phpBB website. The browser tabs include '(1 unread) - rich', 'User risimms loc', 'Cabrillo College', and 'Rich's Cabrillo C'. The address bar shows the URL 'oslab.cis.cabrillo.edu/forum/viewtopic.php?f=101&t=3324&sid=63dda9cf0a544936a540e216474d4c16'. The forum header is blue and contains the phpBB logo, the site title 'Cabrillo College: Computer and Information Systems', and a search bar. Below the header, a breadcrumb trail reads 'Board index < Cabrillo College Fall 2014 Courses < CIS 90 - Fall 2014'. The main content area features a post titled 'Do you need tutoring?' by user 'Takashi Tamasu', dated 'Thu Oct 16, 2014 9:37 pm'. The post text states: 'I belong to the AGS (Alpha Gamma Sigma) Honor Society at Cabrillo and one of the functions this club does is offer FREE tutoring. One of the tutors listed CIS 90 as one of the classes that he is willing to tutor. If someone needs tutoring you can either submit a tutor request form at our site or tell me a number and when you can be reached at that number to arrange tutoring. [https://sites.google.com/site/cabrilloa ... edirects=0](https://sites.google.com/site/cabrilloa...edirects=0)'. The post concludes with 'BTW I am the tutor coordinator for AGS' and 'cheers Takashi'. A user profile box for Takashi Tamasu is visible on the right, showing 'Posts: 59', 'Joined: Wed Jan 29, 2014 3:46 pm', and '5 posts • Page 1 of 1'.



Scripting

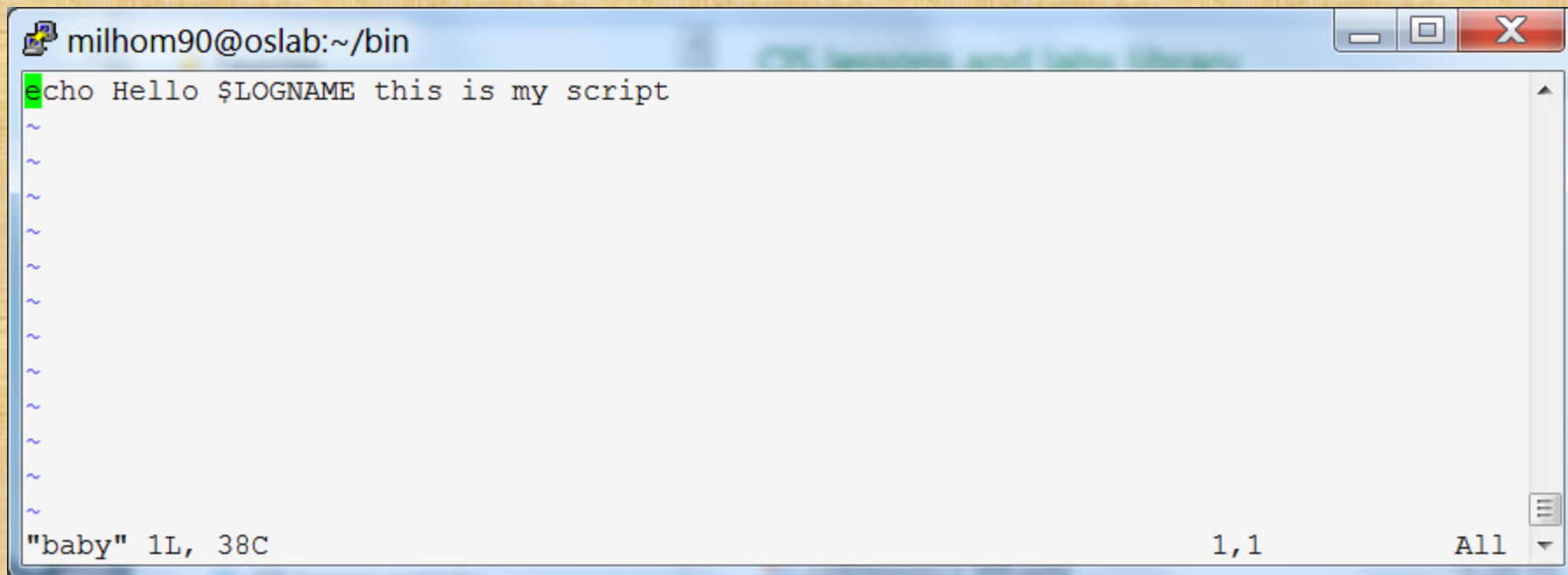


Shell Scripts

- In its simplest form a shell script can just be a list of commands in a file
- Execute "x" permissions must be enabled on the script file.
- The script must either be on your path or you must use an absolute pathname to run it.

Shell Script Examples

```
/home/cis90/milhom/bin $ vim baby
```



A terminal window titled 'milhom90@oslab:~/bin' shows the vim editor in action. The first line of the file 'baby' contains the command 'echo Hello \$LOGNAME this is my script'. The rest of the file is empty, indicated by tilde symbols on subsequent lines. The bottom status bar shows '1,1' and 'All', indicating the cursor is at the beginning of the first line.

use  **:wq** *to save file and quit vi*

```
/home/cis90/milhom/bin $ chmod 750 baby
```

```
/home/cis90/milhom/bin $ baby
```

```
Hello milhom90 this is my script
```

Shell Script Examples

```
/home/cis90/milhom/bin $ vim toddler
```

```
milhom90@oslab:~/bin
#!/bin/bash
# This is a simple script for CIS 90
echo Hello $LOGNAME
date
tty
hostname
exit
~
~
~
~
~
~
-- INSERT --
```

use  **:wq** to save file and quit vi

```
/home/cis90/milhom/bin $ chmod 750 toddler
```

```
/home/cis90/milhom/bin $ toddler
```

```
Hello milhom90
```

```
Mon Nov 25 17:57:15 PST 2013
```

```
/dev/pts/9
```

```
oslab.cishawks.net
```

Shell Script Examples

```
/home/cis90/milhom/bin $ vim dogbone
```

```
milhom90@oslab:~/bin
#!/bin/bash
# Simple interactive script
echo -n "What is your name? "
read NAME
echo -n "What is your favorite bone? "
read BONE
echo "Hi $NAME, your favorite bone is $BONE"
exit 0

~
~
~
~
:wq
```

This is now an interactive script

use  **:wq** to save file and quit vi

```
/home/cis90/milhom/bin $ vim dogbone
```

```
/home/cis90/milhom/bin $ chmod 750 dogbone
```

```
/home/cis90/milhom/bin $ dogbone
```

```
What is your name? Homer
```

```
What is your favorite bone? Turkey
```

```
Hi Homer, your favorite bone is Turkey
```




Final Project myscript

```

milhom90@oslab:~/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
~
    
```

You will modify and extend this script for your final project

Final Project

If you did not do this last week, please do so now

Getting Started

- 1) On Opus, cd to your home directory and enter:
cp ../depot/myscript bin/
- 2) Give your script execute permissions with:
chmod +x bin/myscript
- 3) Run the script:
myscript

Final Project

```
rsimms@oslab:/home/cis90/bin
*****
*           Fall 2014 CIS 90 Online Projects           *
*****
1) Aaron
2) Abraham
3) Alejandrino
4) Ann
5) Benji
6) Cameron
7) Chris
8) Cody
9) Deane
10) Duke
11) Francisco
12) Gabriel
13) Homer
14) James
15) Jeff
16) Jesus
17) Jimmy
18) Jonathan
19) Joshua
20) Justin
21) Luis
22) Matthew
23) Nadia
24) Navin
25) Nick
26) Nicole
27) Paul
28) Richard I.
29) Richard Z.
30) Roberto
31) Ronald
32) Ryan
33) Sam
34) Shea
35) Shenghong
36) Takashi
37) Tommy

99) Exit

Enter Your Choice: █
```

*Before leaving class today you
want to make sure can run
your **myscript** from **allscripts***



Final Project Grading Rubric

Grading rubric (60 points maximum)

Possible Points	Requirements
30	Implementing all five tasks (6 points each): <ul style="list-style-type: none"> • Requirements for each task: <ul style="list-style-type: none"> - Minimum of 10 "original" script command lines - Has comments to explain what it does - Has user interaction
25	You don't have to do all of these but do at least five: <ul style="list-style-type: none"> • Redirecting stdin (5 points) • Redirecting stdout (5 points) • Redirecting stderr (5 points) • Use of permissions (5 points) • Use of filename expansion characters (5 points) • Use of absolute path (5 points) • Use of relative path (5 points) • Use of a PID (5 points) • Use of inodes (5 points) • Use of links (5 points) • Use of a GID or group (5 points) • Use of a UID or user (5 points) • Use of a signal (5 points) • Use of piping (5 points) • Use of an environment variable (5 points) • Use of /bin/mail (5 points) • Use of a conditional (5 points) The maximum for this section are 25 points.
5	Present your script in front of the class
Points lost	
-15	Fails to run from allscripts
-15	Other students in the class are unable to read and execute your script.
-15	Error messages are displayed when running one or more tasks
-up to 90	No credit for any task which contains unoriginal script code that: <ul style="list-style-type: none"> • Doesn't give full credit to the original author • Doesn't indicate where the code was obtained from • Doesn't include licensing terms • Violates copyright or licensing terms
Extra credit	
30	Up to three additional tasks (10 points each)



Final Project


permissions

Permissions

A past forum post ...

Ha Ha Class
Dby on Tue May 12, 2009 12:22 pm

I'm sure this is some kind of payback for last week "Hacking" attempt 😊



```
File Edit View Terminal Help
#!/bin/bash
# menu: A simple menu example
while true
do
clear
echo -n "***** will fail his Final Project"
1) Job 1
2) Task 2
3) Task 3
4) Task 4
5) Task 5
6) Exit
Enter Your Choice:
read RESPONSE
case $RESPONSE in
1) # Complete Task 1
echo "***** will fail his Final Project"
echo "1) Job 1 got hacked!!!!"
echo "what is your name?"
read NAME
echo "what are ur hobbies?"
"myscript" 42L, 646C
```

I will find out who did this 😊😂

~~~~~

ps. Im going to pass 😊

*Uh, oh ... someone got hacked!*



## Group Write Permissions

### ls -l /home/cis90/\*/bin/myscript

```
simben90@oslab:~
/home/cis90/simben $ ls -l /home/cis90/*/bin/myscript
-rwxrwxr-x. 1 albjon90 cis90 714 Nov 18 14:10 /home/cis90/albjon/bin/myscript
-rwxr-xr-x. 1 ayalui90 cis90 690 Nov 18 14:29 /home/cis90/ayalui/bin/myscript
-rwxrwxr-x. 1 bincam90 cis90 764 Nov 18 15:59 /home/cis90/bincam/bin/myscript
-rwxrwxr-x. 1 bownic90 cis90 546 Nov 18 13:51 /home/cis90/bownic/bin/myscript
-rwxrwxr-x. 1 howmil90 cis90 540 Nov 22 13:59 /home/cis90/howmil/bin/myscript
-rwxrwxr-x. 1 isoric90 cis90 704 Nov 18 14:27 /home/cis90/isoric/bin/myscript
-rwxrwxr-x. 1 keichr90 cis90 546 Nov 18 13:53 /home/cis90/keichr/bin/myscript
-rwxrwxr-x. 1 lamnav90 cis90 790 Nov 18 14:13 /home/cis90/lamnav/bin/myscript
-rwxrwxr-x. 1 lishe90 cis90 790 Nov 18 14:11 /home/cis90/lishe/bin/myscript
-rwxrwxr-x. 1 locaar90 cis90 692 Nov 18 14:12 /home/cis90/locaar/bin/myscript
-rwxrwxr-x. 1 milhom90 cis90 794 Nov 20 14:53 /home/cis90/milhom/bin/myscript
-rwxrwxr-x. 1 nordak90 cis90 716 Nov 18 14:12 /home/cis90/nordak/bin/myscript
-rwxrwxr-x. 1 pikann90 cis90 713 Nov 24 10:49 /home/cis90/pikann/bin/myscript
-rwxrwxr-x. 1 porrya90 cis90 815 Nov 20 14:51 /home/cis90/porrya/bin/myscript
-rwxrwxr-x. 1 rodjus90 cis90 721 Nov 18 14:13 /home/cis90/rodjus/bin/myscript
-rwxrwxr-x. 1 simben90 cis90 546 Nov 17 15:53 /home/cis90/simben/bin/myscript
-rwxr-x--x. 1 smimat90 cis90 546 Nov 24 10:12 /home/cis90/smimat/bin/myscript
-rwxrwxr-x. 1 tamjim90 cis90 713 Nov 18 14:41 /home/cis90/tamjim/bin/myscript
-rwxrwxr-x. 1 tamtak90 cis90 796 Nov 18 14:18 /home/cis90/tamtak/bin/myscript
-rwxrwx--x. 1 urijes90 cis90 519 Nov 24 18:01 /home/cis90/urijes/bin/myscript
-rwxrwxr-x. 1 wrenic90 cis90 703 Nov 18 14:44 /home/cis90/wrenic/bin/myscript
-rwxrwxr-x. 1 zemric90 cis90 734 Nov 20 13:08 /home/cis90/zemric/bin/myscript
/home/cis90/simben $
```

Which **myscript** files can only be edited by their owner?  
 Which ones could be edited by anyone in the CIS 90 class?  
 Which ones could be edited by anyone on Opus?

## Group Read and Execute Permissions

```

rsimms@oslab:~$ /home/cis90/bin/checkmyscripts
ls: cannot access /home/cis90/diljam/bin/myscript: No such file or directory
-rwxrwxr-x. 1 locaar90 cis90 692 Nov 18 14:12 /home/cis90/locaar/bin/myscript
ls: cannot access /home/cis90/tranad/bin/myscript: No such file or directory
ls: cannot access /home/cis90/nieabr/bin/myscript: No such file or directory
-rwxrwxr-x. 1 bownic90 cis90 546 Nov 18 13:51 /home/cis90/bownic/bin/myscript
ls: cannot access /home/cis90/boyjef/bin/myscript: No such file or directory
ls: cannot access /home/cis90/dobtho/bin/myscript: No such file or directory
ls: cannot access /home/cis90/espale/bin/myscript: No such file or directory
-rwxrwxr-x. 1 pikann90 cis90 713 Nov 24 10:49 /home/cis90/pikann/bin/myscript
ls: cannot access /home/cis90/quifra/bin/myscript: Permission denied
-rwxrwxr-x. 1 rodjus90 cis90 721 Nov 18 14:13 /home/cis90/rodjus/bin/myscript
-rwxrwxr-x. 1 tamjim90 cis90 713 Nov 18 14:41 /home/cis90/tamjim/bin/myscript
-rwxrwxr-x. 1 tamtak90 cis90 796 Nov 18 14:18 /home/cis90/tamtak/bin/myscript
-rwxrwx--x. 1 urijes90 cis90 519 Nov 24 18:01 /home/cis90/urijes/bin/myscript
-rwxrwxr-x. 1 wrenic90 cis90 703 Nov 18 14:44 /home/cis90/wrenic/bin/myscript
ls: cannot access /home/cis90/zahpau/bin/myscript: No such file or directory
-rwxrwxr-x. 1 zemric90 cis90 734 Nov 20 13:08 /home/cis90/zemric/bin/myscript
-rwxrwxr-x. 1 howmil90 cis90 540 Nov 22 13:59 /home/cis90/howmil/bin/myscript
-rwxrwxr-x. 1 albjon90 cis90 714 Nov 18 14:10 /home/cis90/albjon/bin/myscript
ls: cannot access /home/cis90/asngab/bin/myscript: No such file or directory
ls: cannot access /home/cis90/atirob/bin/myscript: No such file or directory
-rwxr-xr-x. 1 ayalui90 cis90 690 Nov 18 14:29 /home/cis90/ayalui/bin/myscript
-rwxrwxr-x. 1 bincam90 cis90 764 Nov 18 15:59 /home/cis90/bincam/bin/myscript
ls: cannot access /home/cis90/desmat/bin/myscript: No such file or directory
-rwxrwxr-x. 1 isoric90 cis90 704 Nov 18 14:27 /home/cis90/isoric/bin/myscript
-rwxrwxr-x. 1 keichr90 cis90 546 Nov 18 13:53 /home/cis90/keichr/bin/myscript
-rwxrwxr-x. 1 lamnav90 cis90 790 Nov 18 14:13 /home/cis90/lamnav/bin/myscript
-rwxrwxr-x. 1 lishe90 cis90 790 Nov 18 14:11 /home/cis90/lishe/bin/myscript
-rwxrwxr-x. 1 nordak90 cis90 716 Nov 18 14:12 /home/cis90/nordak/bin/myscript
-rwxrwxr-x. 1 porrya90 cis90 815 Nov 20 14:51 /home/cis90/porrya/bin/myscript
ls: cannot access /home/cis90/specod/bin/myscript: No such file or directory
ls: cannot access /home/cis90/tinsam/bin/myscript: No such file or directory
rsimms@oslab ~]$

```

*Which myscrip files cannot be run by classmates?*

## Class Activity

Note: One of the requirements for the final project is setting permissions on your script so that all cis90 members can read and run it.

To meet this requirement use:

```
cd  
chmod 750 bin bin/myscript  
ls -ld bin bin/myscript
```

umask  
again!

# Permissions

Why can other classmates write to my scripts?

## *Before Lab 10*

```
/home/cis90/simben/bin $ umask
0002
/home/cis90/simben $ rm newscript; touch newscript
/home/cis90/simben $ ls -l newscript
-rw-rw-r-- 1 simben cis90 0 Nov 23 16:17 newscript
/home/cis90/simben $ chmod +x newscript
/home/cis90/simben $ ls -l newscript
-rwxrwxr-x 1 simben cis90 0 Nov 23 16:17 newscript
```

## *After Lab 10*

```
/home/cis90/simben $ umask
0006
/home/cis90/simben $ rm newscript; touch newscript
/home/cis90/simben $ ls -l newscript
-rw-rw---- 1 simben cis90 0 May 12 08:44 newscript
/home/cis90/simben $ chmod +x newscript
/home/cis90/simben $ ls -l newscript
-rwxrwx--x 1 simben cis90 0 May 12 08:44 newscript
```

*Because your umask setting allows group members to have write permission on any new files you create!*

# Permissions

```
[rodduk90@opus bin]$ cat /home/cis90/rodduk/.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
```

```
eval `tset -s -m vt100:vt100 -m :\?${TERM:-ansi} -r -Q `
```

*Note your umask is defined in .bash\_profile which runs every time you login. In lab 10 you change this setting to 006.*



## Class Activity

- Change your umask to 026
- Can group or other users modify future new files now?
- Try it, **touch** a new file and check the permissions with **ls -l**
- How would you make this a permanent umask setting?

# Final Project Getting Started

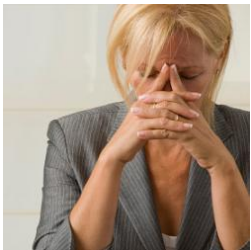


# What takes longer?



**Writing the script?**

**Or deciding what to script?**



One way to get started ... select a random command to build a script around

### Commands

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



*For this example we will pick the grep command*

# Research your command by reading the man page and googling examples

```

rsimms@opus:~/cis90/project
GREP (1)
NAME
    grep, egrep, fgrep - print lines matching a pattern

SYNOPSIS
    grep [options] PATTERN [FILE...]
    grep [options] [-e PATTERN | -f FILE] [FILE...]

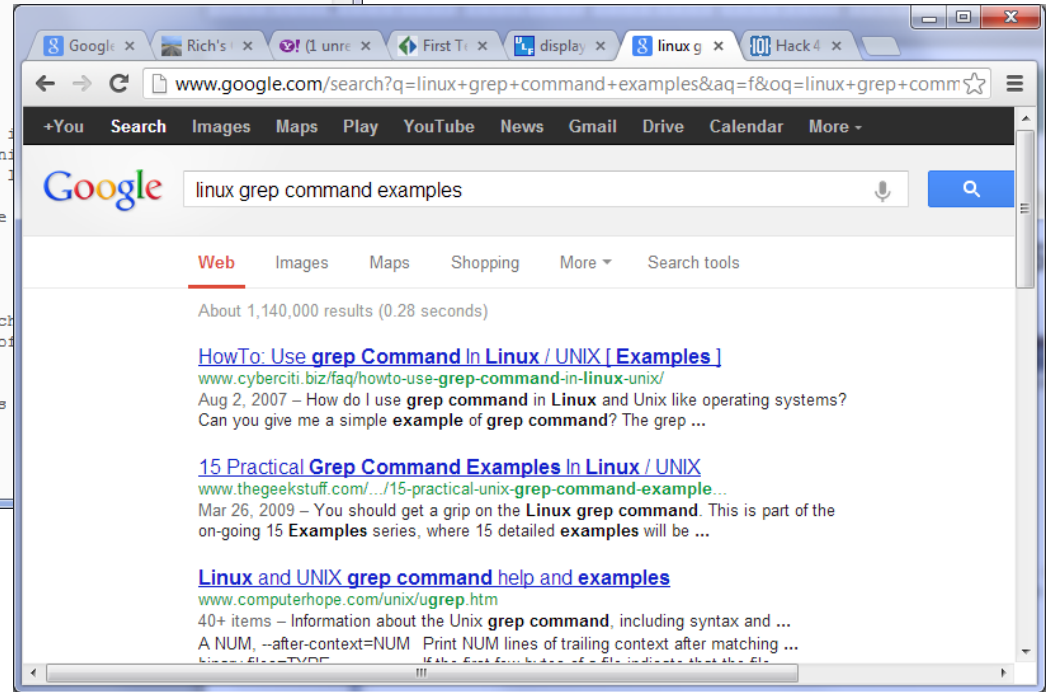
DESCRIPTION
    Grep searches the named input FILES (or standard input, if no files are
    named, or the file name - is given) for lines containing the pattern
    given PATTERN. By default, grep prints the matching lines.

    In addition, two variant programs egrep and fgrep are provided. Egrep
    is the same as grep -E. Fgrep is the same as grep -F.

OPTIONS
    -A NUM, --after-context=NUM
        Print NUM lines of trailing context after matching lines. This
        option is only valid when the -n option is also specified.

    -a, --text
        Process a binary file as if it were text; this option is only
        valid when used with -E, -F, or -R.

    -B NUM, --before-context=NUM
        Print NUM lines of leading context before matching lines. This
        option is only valid when the -n option is also specified.
    
```



*Review the various options and arguments for the command*

Next, decide what you want to do with the command you selected. For this example we will:

1. Start a new task in **myscript**
2. Customize the menu for the new task
3. Start with a simple **grep** command
4. Add some simple interaction
5. Add successive grep commands that experiment with different options
6. Iterate till happy with it.

## Start hacking the menu!

*Customize the menu options for Task 1*

*After*

```
rodduk90@oslab:~/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
            ;;
        *)
            ;;
    esac
done
"myscript" 37L, 546C
```

*Before*

```
rodduk90@oslab:~/bin
#!/bin/bash
#
# menu: A simple menu template
#
while true
do
    clear
    echo -n "
        CIS 90 Final Project
    1) Hacking with the grep command
    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
            ;;
        *)
            ;;
    esac
done
-- INSERT --
10,5-12 Top
```

← → C www.catb.org/jargon/html/H/hacker.html

**hacker:** n.

[originally, someone who makes furniture with an axe]

1. A person who enjoys exploring the details of programmable systems and how to stretch their capabilities, as opposed to most users, who prefer to learn only the minimum necessary. RFC1392, the *Internet Users' Glossary*, usefully amplifies this as: A person who delights in having an intimate understanding of the internal workings of a system, computers and computer networks in particular.
2. One who programs enthusiastically (even obsessively) or who enjoys programming rather than just theorizing about programming.
3. A person capable of appreciating [hack value](#).
4. A person who is good at programming quickly.
5. An expert at a particular program, or one who frequently does work using it or on it; as in 'a Unix hacker'. (Definitions 1 through 5 are correlated, and people who fit them congregate.)
6. An expert or enthusiast of any kind. One might be an astronomy hacker, for example.
7. One who enjoys the intellectual challenge of creatively overcoming or circumventing limitations.
8. [deprecated] A malicious meddler who tries to discover sensitive information by poking around. Hence password hacker, network hacker. The correct term for this sense is [cracker](#).

The term 'hacker' also tends to connote membership in the global community defined by the net (see [the network](#). For discussion of some of the basics of this culture, see the [How To Become A Hacker](#) FAQ. It also implies that the person described is seen to subscribe to some version of the hacker ethic (see [hacker ethic](#)).

It is better to be described as a hacker by others than to describe oneself that way. Hackers consider themselves something of an elite (a meritocracy based on ability), though one to which new members are gladly welcome. There is thus a certain ego satisfaction to be had in identifying yourself as a hacker (but if you claim to be one and are not, you'll quickly be labeled [bogus](#)). See also [geek](#), [wannabee](#).

This term seems to have been first adopted as a badge in the 1960s by the hacker culture surrounding TMRC and the MIT AI Lab. We have a report that it was used in a sense close to this entry's by teenage radio hams and electronics tinkerers in the mid-1950s.

*Hacking (building, exploring) is not cracking (malicious)*

# Layout your work area on the screen

```

rodduk90@oslab:~/bin
#!/bin/bash
#
# menu: A simple menu template
#
while true
do
    clear
    echo -n "
        CIS 90 Final Project
    1) Hacking with the grep command
    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
~
~
~
-- INSERT --
1,12 All
    
```

1st

```

rodduk90@oslab:~/bin
/home/cis90/rodduk $ cd bin
/home/cis90/rodduk/bin $ myscript
    
```

2nd

```

rodduk90@oslab:~
GREP(1)
NAME
    grep, egrep, fgrep - print lines matching a pattern

SYNOPSIS
    grep [OPTIONS] PATTERN [FILE...]
    grep [OPTIONS] [-e PATTERN | -f FILE] [FILE...]

DESCRIPTION
    grep searches the named input FILES (or standard input if no files are
    named, or if a single hyphen-minus (-) is given as file name) for lines
    containing a match to the given PATTERN. By default, grep prints the
    matching lines.

    In addition, two variant programs egrep and fgrep are available. egrep
    is the same as grep -E. fgrep is the same as grep -F. Direct
    invocation as either egrep or fgrep is deprecated, but is provided to
    allow historical applications that rely on them to run unmodified.

OPTIONS
    Generic Program Information
    --help Print a usage message briefly summarizing these command-line
    :
    
```

3rd

Utilize screen real estate with multiple windows:

- the 1<sup>st</sup> for vi,
- the 2<sup>nd</sup> for testing **myscript**,
- and a 3<sup>rd</sup> for experimenting or showing man pages

# Test your menu change

```

rodduk90@oslab:~/bin
#!/bin/bash
#
# menu: A simple menu template
#
while true
do
    clear
    echo -n "
        CIS 90 Final Project
    1) Hacking with the grep command
    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
~
~
~
"myscript" 37L, 569C written          1,11          All
    
```

```

rodduk90@oslab:~/bin
        CIS 90 Final Project
    1) Hacking with the grep command
    2) Task 2
    3) Task 3
    4) Task 4
    5) Task 5
    6) Exit

    Enter Your Choice: █
    
```

*Changes work!*

```

rodduk90@oslab:~
GREP(1)                                GREP(1)
NAME
    grep, egrep, fgrep - print lines matching a pattern

SYNOPSIS
    grep [OPTIONS] PATTERN [FILE...]
    grep [OPTIONS] [-e PATTERN | -f FILE] [FILE...]

DESCRIPTION
    grep searches the named input FILEs (or standard input if no files are
    named, or if a single hyphen-minus (-) is given as file name) for lines
    containing a match to the given PATTERN. By default, grep prints the
    matching lines.

    In addition, two variant programs egrep and fgrep are available. egrep
    is the same as grep -E. fgrep is the same as grep -F. Direct
    invocation as either egrep or fgrep is deprecated, but is provided to
    allow historical applications that rely on them to run unmodified.

OPTIONS
    Generic Program Information
    --help Print a usage message briefly summarizing these command-line
    :
    
```

Run **myscript** in the 2<sup>nd</sup> window and verify your changes work



# Find the location to insert your new task commands

```

rodduk90@oslab:~/bin
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
-- INSERT --
12,5-12 78%

```

*Now its time to add some commands to the task.*

*Be sure to insert commands **after** the generic comment and **before** the ;;*

# Add a simple command first and test it

```

rodduk90@oslab:~/bin
#!/bin/bash
#
# menu: A simple menu template
#
while true
do
    clear
    echo -n "
        CIS 90 Final Project
    1) Hacking with the grep command
    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
            grep beauty poems/**
            ;;
        2) # Commands for Task 2
            ;;
        3) # Commands for Task 3
            ;;
        4) # Commands for Task 4
            ;;
        5) # Commands for Task 5
            ;;
        *) echo "Please enter a number between 1 and 6"
            ;;
    esac
    echo -n "Hit the Enter key to return to menu "
    read dummy
done
~
~
"myscript" 38L, 593C written          21,15-29    All
    
```

```

rodduk90@oslab:~/bin

        CIS 90 Final Project
    1) Hacking with the grep command
    2) Task 2
    3) Task 3
    4) Task 4
    5) Task 5
    6) Exit

    Enter Your Choice: 1
grep: poems/**: No such file or directory
Hit the Enter key to return to menu █
    
```

 *Oops, the change broke the script! Why? Because the relative path (beauty poems/\*\*) does not work from the bin directory*

```

rodduk90@oslab:~/
/home/cis90/rodduk $ grep beauty poems/**
poems/Shakespeare/sonnet1:That thereby beauty's rose might never die,
poems/Shakespeare/sonnet10: That beauty still may live in thine or thee.
poems/Shakespeare/sonnet11:Herein lives wisdom, beauty, and increase;
poems/Shakespeare/sonnet17:If I could write the beauty of your eyes,
poems/Shakespeare/sonnet2:And dig deep trenches in thy beauty's field,
poems/Shakespeare/sonnet2:Then being ask'd, where all thy beauty lies,
poems/Shakespeare/sonnet2:How much more praise deserv'd thy beauty's use,
poems/Shakespeare/sonnet2:Proving his beauty by succession thine.
poems/Shakespeare/sonnet4:Upon thyself thy beauty's legacy?
poems/Shakespeare/sonnet4: Thy unus'd beauty must be tomb'd with thee,
poems/Shakespeare/sonnet5:Beauty's effect with beauty were bereft,
poems/Shakespeare/sonnet7:Yet mortal looks adore his beauty still,
poems/Shakespeare/sonnet9:But beauty's waste hath in the world an end,
poems/Yeats/old:And loved your beauty with love false or true,
/home/cis90/rodduk $ █
    
```

Experiment with a **grep** command in 3<sup>rd</sup> window

In the 1<sup>st</sup> window add the new grep command then save with **<esc>:w** (don't quit vi)

Run **myscript** in the 2<sup>nd</sup> second window to test change.

## Fix it and test again

```

rodduk90@oslab:~/bin
#!/bin/bash
#
# menu: A simple menu template
#
while true
do
    clear
    echo -n "
        CIS 90 Final Project
    1) Hacking with the grep command
    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
            grep beauty /home/cis90/rodduk/poems/*/*
            ;;
        2) # Commands for Task 2
            ;;
        3) # Commands for Task 3
            ;;
        4) # Commands for Task 4
            ;;
        5) # Commands for Task 5
            ;;
        *) echo "Please enter a number between 1 and 6"
            ;;
    esac
    echo -n "Hit the Enter key to return to menu "
    read dummy
done
~
~
"myscript" 38L, 612C written                21,33-47    All
    
```

```

rodduk90@oslab:~/bin
        CIS 90 Final Project
    1) Hacking with the grep command
    2) Task 2
    3) Task 3
    4) Task 4
    5) Task 5
    6) Exit

    Enter Your Choice: 1
/home/cis90/rodduk/poems/Shakespeare/sonnet1:That thereby beauty's rose might ne
ver die,
/home/cis90/rodduk/poems/Shakespeare/sonnet10: That beauty still may live in th
ine or thee.
/home/cis90/rodduk/poems/Shakespeare/sonnet11:Herein lives wisdom, beauty, and i
ncrease;
/home/cis90/rodduk/poems/Shakespeare/sonnet17:If I could write the beauty of you
r eyes,
/home/cis90/rodduk/poems/Shakespeare/sonnet2:And dig deep trenches in thy beauty
's field,
/home/cis90/rodduk/poems/Shakespeare/sonnet2:Then being ask'd, where all thy bea
uty lies,
/home/cis90/rodduk/poems/Shakespeare/sonnet2:How much more praise deserv'd thy b
eauty's use,
/home/cis90/rodduk/poems/Shakespeare/sonnet2:Proving his beauty by succession th
ine.
/home/cis90/rodduk/poems/Shakespeare/sonnet4:Upon thyself thy beauty's legacy?
/home/cis90/rodduk/poems/Shakespeare/sonnet4: Thy unus'd beauty must be tomb'd
with thee,
/home/cis90/rodduk/poems/Shakespeare/sonnet5:Beauty's effect with beauty were be
reft,
/home/cis90/rodduk/poems/Shakespeare/sonnet7:Yet mortal looks adore his beauty s
till,
/home/cis90/rodduk/poems/Shakespeare/sonnet9:But beauty's waste hath in the worl
d an end,
/home/cis90/rodduk/poems/Yeats/old:And loved your beauty with love false or true
,
Hit the Enter key to return to menu
    
```

Fix worked! 😊

Fix task in 1<sup>st</sup> window by using an absolute pathname then save with **<esc>:w**

Re-run **myscript** in the 2<sup>nd</sup> second window and test your change. To do this quickly hit **Ctrl-C** then **<up arrow>** key.

```

/home/cis90/rodduk/poems/Shakespeare/sonnet5:Beauty's effect with beauty were bereft,
/home/cis90/rodduk/poems/Shakespeare/sonnet7:Yet mortal looks adore his beauty still,
/home/cis90/rodduk/poems/Shakespeare/sonnet9:But beauty's waste hath in the world an end,
/home/cis90/rodduk/poems/Yeats/old:And loved your beauty with love false or true,
/home/cis90/rodduk $
    
```

## Add some interaction

```

rodduk90@oslab:~/bin
#!/bin/bash
#
# menu: A simple menu template
#
while true
do
    clear
    echo -n "
    CIS 90 Final Project
    1) Hacking with the grep command
    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
            echo "Are you ready to search for beauty in the poems?"
            read dummy
            grep beauty /home/cis90/rodduk/poems/*/*
            ;;
        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
"myscript" 40L, 711C written

```

*Let's add some interaction*

1) # Commands for Task 1

`echo "Are you ready to search for beauty in the poems?"`

`read dummy`

`grep beauty /home/cis90/rodduk/poems/*/*`

`;;`

```

rodduk90@oslab:~/bin
CIS 90 Final Project
1) Hacking with the grep command
2) Task 2
3) Task 3
4) Task 4
5) Task 5
6) Exit

Enter Your Choice: 1
Are you ready to search for beauty in the poems?

/home/cis90/rodduk/poems/Shakespeare/sonnet1:That thereby beauty's rose might never die,
/home/cis90/rodduk/poems/Shakespeare/sonnet10: That beauty still may live in thine or thee.
/home/cis90/rodduk/poems/Shakespeare/sonnet11:Herein lives wisdom, beauty, and increase;
/home/cis90/rodduk/poems/Shakespeare/sonnet17:If I could write the beauty of your eyes,
/home/cis90/rodduk/poems/Shakespeare/sonnet2:And dig deep trenches in thy beauty's field,
/home/cis90/rodduk/poems/Shakespeare/sonnet2:Then being ask'd, where all thy beauty lies,
/home/cis90/rodduk/poems/Shakespeare/sonnet2:How much more praise deserv'd thy beauty's use,
/home/cis90/rodduk/poems/Shakespeare/sonnet2:Proving his beauty by succession thine.
/home/cis90/rodduk/poems/Shakespeare/sonnet4:Upon thyself thy beauty's legacy?
/home/cis90/rodduk/poems/Shakespeare/sonnet4: Thy unus'd beauty must be tomb'd with thee,
/home/cis90/rodduk/poems/Shakespeare/sonnet5:Beauty's effect with beauty were bereft,
/home/cis90/rodduk/poems/Shakespeare/sonnet7:Yet mortal looks adore his beauty still,
/home/cis90/rodduk/poems/Shakespeare/sonnet9:But beauty's waste hath in the world an end,
/home/cis90/rodduk/poems/Yeats/old:And loved your beauty with love false or true,
Hit the Enter key to return to menu

```

*And it works!*

# Try a new option on the command

```

rodduk90@oslab:~/bin
#!/bin/bash
#
# menu: A simple menu template
#
while true
do
    clear
    echo -n "
    CIS 90
    1) Hacking with the grep command
    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
            echo "Are you ready to search for beauty in the poems?"
            read dummy
            grep -h beauty /home/cis90/rodduk/poems/*/*
            ;;
        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
"myscript" 40L, 714C written
    
```

*Let's try the -h option and not print the leading file names*

```

1) # Commands for Task 1
echo "Are you ready to search for beauty in the poems?"
read dummy
grep -h beauty /home/cis90/rodduk/poems/*/*
;;
    
```

```

rodduk90@oslab:~/bin
CIS 90 Final Project
1) Hacking with the grep command
2) Task 2
3) Task 3
4) Task 4
5) Task 5
6) Exit

Enter Your Choice: 1
Are you ready to search for beauty in the poems?
1
That thereby beauty's rose might never die,
    That beauty still may live in thine or thee.
Herein lives wisdom, beauty, and increase;
If I could write the beauty of your eyes,
And dig deep trenches in thy beauty's field,
Then being ask'd, where all thy beauty lies,
How much more praise deserv'd thy beauty's use,
Proving his beauty by succession thine.
Upon thyself thy beauty's legacy?
    Thy unus'd beauty must be tomb'd with thee,
Beauty's effect with beauty were bereft,
Yet mortal looks adore his beauty still,
But beauty's waste hath in the world an end,
And loved your beauty with love false or true,
Hit the Enter key to return to menu
    
```

*And it works!*

# Add a new feature

Let's add a count of the strings found now

1) # Commands for Task 1

```

echo "Are you ready to search for beauty in the poems?"
read dummy
grep -h beauty /home/cis90/rodduk/poems/*/*
echo "Ready to count them?"
read dummy
grep -h beauty /home/cis90/rodduk/poems/*/* | wc -l

```

```

case $RESPONSE in
1) # Commands for Task 1
echo "Are you ready to search for beauty in the poems?"
read dummy
grep -h beauty /home/cis90/rodduk/poems/*/*
echo "Ready to count them?"
read dummy
grep -h beauty /home/cis90/rodduk/poems/*/* | wc -l
;;
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

```

"myscript" 43L, 839C written 26, 53-67

```

CIS 90 Final Project
1) Hacking with the grep command
2) Task 2
3) Task 3
4) Task 4
5) Task 5
6) Exit

Enter Your Choice: 1
Are you ready to search for beauty in the poems?

That thereby beauty's rose might never die,
That beauty still may live in thine or thee.
Herein lives wisdom, beauty, and increase;
If I could write the beauty of your eyes,
And dig deep trenches in thy beauty's field,
Then being ask'd, where all thy beauty lies,
How much more praise deserv'd thy beauty's use,
Proving his beauty by succession thine.
Upon thyself thy beauty's legacy?
Thy unus'd beauty must be tomb'd with thee,
Beauty's effect with beauty were bereft,
Yet mortal looks adore his beauty still,
But beauty's waste hath in the world an end,
And loved your beauty with love false or true,
Ready to count them?

14
Hit the Enter key to return to menu

```

## How many points so far?

*Let's score our mini-script so far*

```
1) # Commands for Task 1
echo "Are you ready to search for beauty in the poems?"
read dummy
grep -h beauty /home/cis90/rodduk/poems/*/*
echo "Ready to count them?"
read dummy
grep -h beauty /home/cis90/rodduk/po
;;
```

Implementing all five tasks (6 points each):

- Requirements for each task:
- NO** -Minimum of 10 "original" script command lines
- NO** -Has one or more non-generic comments to explain what it is doing
- ✓ -Has user interaction

You don't have to do all of these but do at least five:

- Redirecting stdin (5 points)
- Redirecting stdout (5 points)
- Redirecting stderr (5 points)
- Use of permissions (5 points)
- ✓ • Use of filename expansion characters (5 points)
- ✓ • Use of absolute path (5 points)
- Use of relative path (5 points)
- Use of a PID (5 points)
- Use of inodes (5 points)
- Use of links (5 points)
- Use of scheduling (5 points)
- Use of a GID or group (5 points)
- Use of a UID or user (5 points)
- Use of a /dev/tty device (5 points)
- Use of a signal (5 points)
- ✓ • Use of piping (5 points)
- Use of an environment variable (5 points)
- Use of /bin/mail (5 points)
- Use of a conditional (5 points)

The maximum for this section is 25 points.

# Make another enhancement

*Enhance script to let user specify search string and use color*

```

1) # Commands for Task 1
echo "Are you ready to search for beauty in the poems?"
read dummy
grep -h beauty /home/cis90/rodduk/poems/*/*
echo "Ready to count them?"
read dummy
2)
3) grep -h beauty /home/cis90/rodduk/poems/*/* | wc -l
4) echo "Enter a new string to search for"
5) read string
6) echo searching for '$string'
grep -h --color $string /home/cis90/rodduk/poems/*/*

;;

```

```

read dummy
grep -h beauty /home/cis90/rodduk/poems/*/*
echo "Ready to count them?"
read dummy
grep -h beauty /home/cis90/rodduk/poems/*/* | wc -l
echo "Enter a new string to search for"
read string
echo searching for '$string'
grep -h --color $string /home/cis90/rodduk/poems/*/*
;;

```

```

rodduk90@oslab:~/bin
5) Task 5
6) Exit

Enter Your Choice: 1
Are you ready to search for beauty in the poems?

That thereby beauty's rose might never die,
That beauty still may live in thine or thee.
Herein lives wisdom, beauty, and increase;
If I could write the beauty of your eyes,
And dig deep trenches in thy beauty's field,
Then being ask'd, where all thy beauty lies,
How much more praise deserv'd thy beauty's use,
Proving his beauty by succession thine.
Upon thyself thy beauty's legacy?
Thy unus'd beauty must be tomb'd with thee,
Beauty's effect with beauty were bereft,
Yet mortal looks adore his beauty still,
But beauty's waste hath in the world an end,
And loved your beauty with love false
Ready to count them?

14
Enter a new string to search for
sweet
searching for "sweet"
Thyself thy foe, to thy sweet self too cruel.
To show me worthy of thy sweet respect:
To thy sweet will making addition thus.
Thou of thyself thy sweet self dost deceive,
Leese but their show, their substance still lives sweet.
Hit the Enter key to return to menu

```

*And it works!*



## Check the score again

### Let's re-score modified script

```
1) # Commands for Task 1
echo "Are you ready to search for beauty in the poems?"
read dummy
grep -h beauty /home/cis90/rodduk/poems/*/*
echo "Ready to count them?"
read dummy
grep -h beauty /home/cis90/rodduk/poems/*/*
echo "Enter a new string to search for:"
read string
echo searching for "'$string'"
grep -h --color $string /home/cis90/rodduk/poems/*/*
;;
```

#### Implementing all five tasks (6 points each):

- Requirements for each task:
  - ✓ -Minimum of 10 "original" script command lines
  - NO -Has one or more non-generic comments to explain what it is doing
  - ✓ -Has user interaction

#### You don't have to do all of these but do at least five:

- Redirecting stdin (5 points)
- Redirecting stdout (5 points)
- Redirecting stderr (5 points)
- Use of permissions (5 points)
- ✓ • Use of filename expansion characters (5 points)
- ✓ • Use of absolute path (5 points)
- Use of relative path (5 points)
- Use of a PID (5 points)
- Use of inodes (5 points)
- Use of links (5 points)
- Use of scheduling (5 points)
- Use of a GID or group (5 points)
- Use of a UID or user (5 points)
- Use of a /dev/tty device (5 points)
- Use of a signal (5 points)
- ✓ • Use of piping (5 points)
- Use of an environment variable (5 points)
- Use of /bin/mail (5 points)
- Use of a conditional (5 points)

The maximum for this section is 25 points.

# Bing - one task done that meets minimum requirements!

*Add some comments to help others understand what you are doing*

```
1) # Task 1 - grep command explored

# Simple grep for "beauty"
echo "Are you ready to search for beauty in the poems?"
read dummy
grep -h beauty /home/cis90/rodduk/poem

# Same as before but counts matches to
echo "Ready to count them?"
read dummy
grep -h beauty /home/cis90/rodduk/poem

# Prompt user to supply search string
echo "Enter a new string to search for"
read string
echo searching for "'$string'"
grep -h $string /home/cis90/rodduk/poem
;;
```

Implementing all five tasks (6 points each):

- Requirements for each task:
  - ✓ -Minimum of 10 "original" script command lines
  - ✓ -Has one or more non-generic comments to explain what it is doing
  - ✓ -Has user interaction

You don't have to do all of these but do at least five:

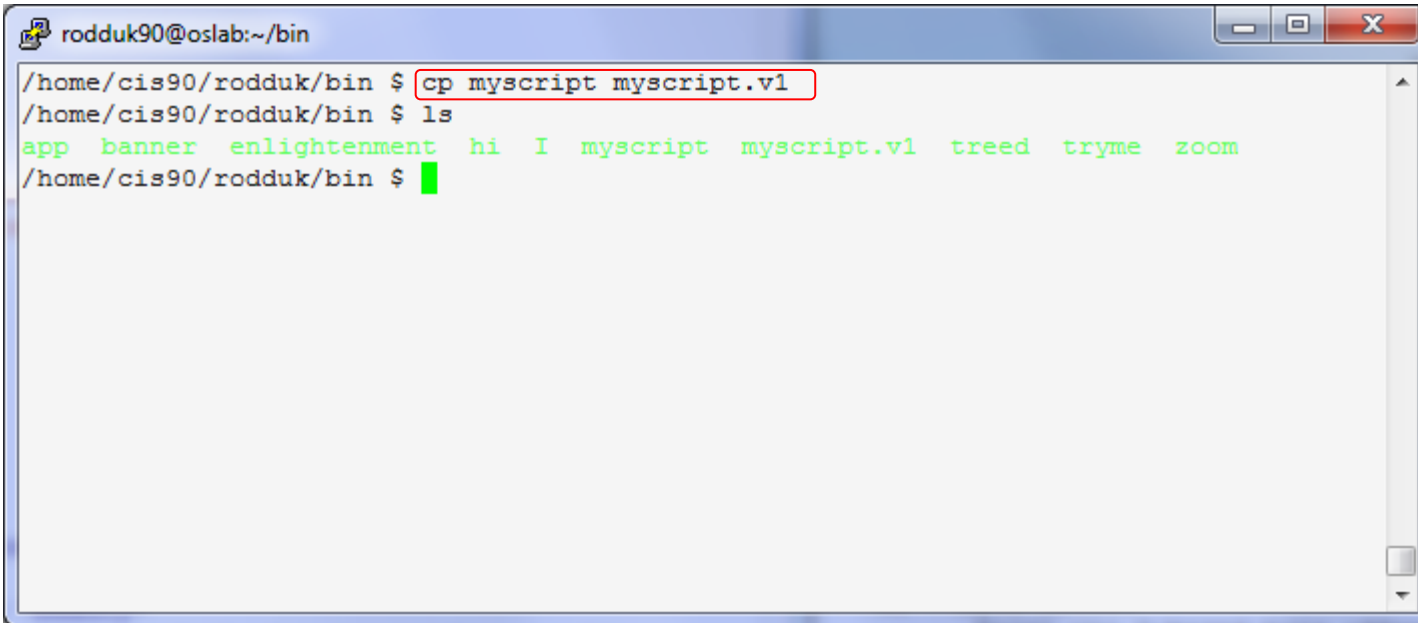
- Redirecting stdin (5 points)
- Redirecting stdout (5 points)
- Redirecting stderr (5 points)
- Use of permissions (5 points)
- ✓ Use of filename expansion characters (5 points)
- ✓ Use of absolute path (5 points)
- Use of relative path (5 points)
- Use of a PID (5 points)
- Use of inodes (5 points)
- Use of links (5 points)
- Use of scheduling (5 points)
- Use of a GID or group (5 points)
- Use of a UID or user (5 points)
- Use of a /dev/tty device (5 points)
- Use of a signal (5 points)
- ✓ Use of piping (5 points)
- Use of an environment variable (5 points)
- Use of /bin/mail (5 points)
- Use of a conditional (5 points)

The maximum for this section is 25 points.

*And has fulfilled three of the five requirements for the overall project!*

## Backup your work!

`cp myscript myscript.v1` *after first day of work*



```
rodduk90@oslab:~/bin
/home/cis90/rodduk/bin $ cp myscript myscript.v1
/home/cis90/rodduk/bin $ ls
app banner enlightenment hi I myscript myscript.v1 treed tryme zoom
/home/cis90/rodduk/bin $
```

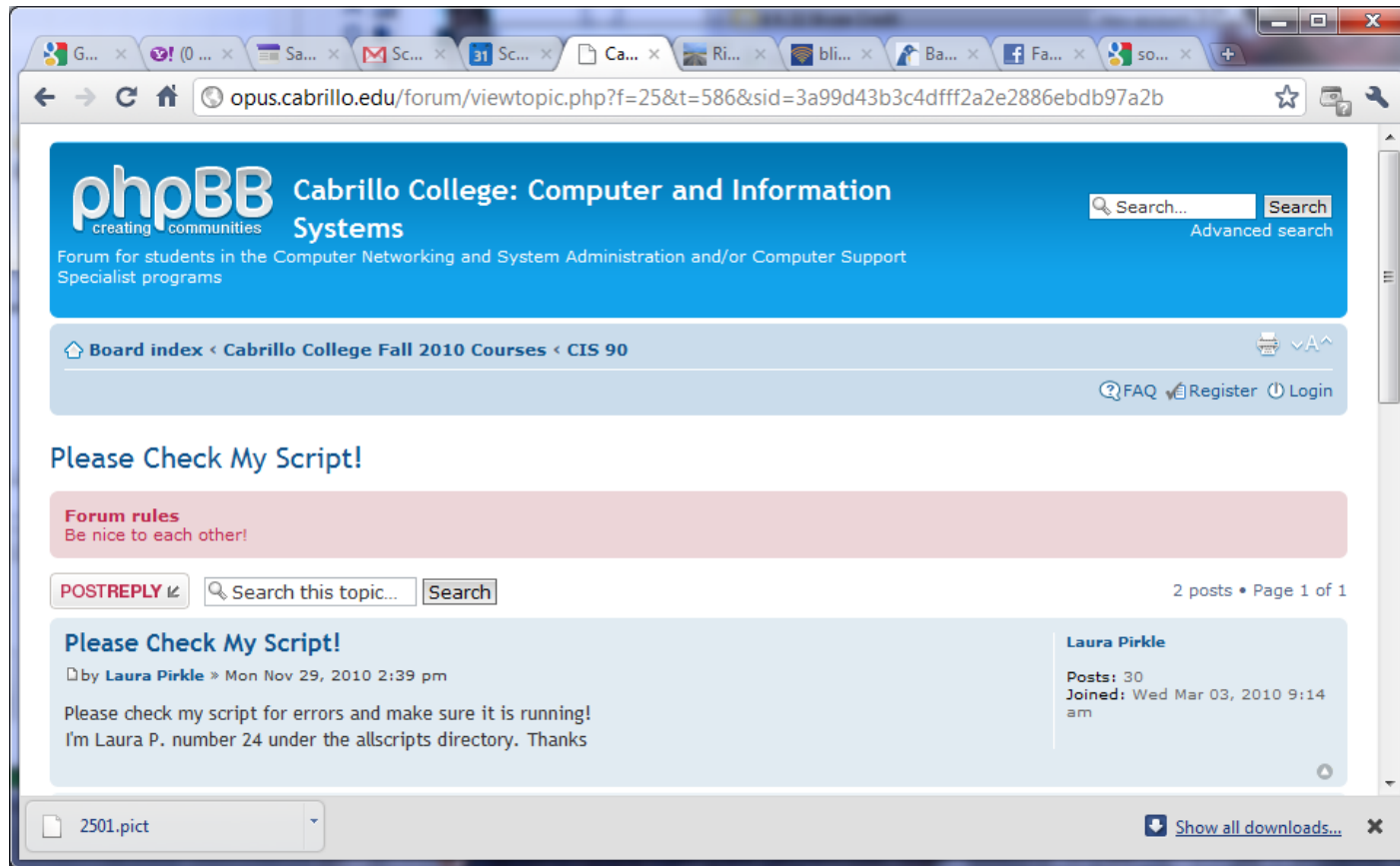
`cp myscript myscript.v2` *after second day of work*

`cp myscript myscript.v3` *and so on ...*

`cp myscript myscript.v4`

*Always be able to revert back to an earlier version in case you clobber the current one!*

# Testing your script



*The ask others on the forum to check your script and give you feedback*

## Plan extra time for:

- Figuring out how to do what you really want to do!
- Removing syntax errors
- Removing logic errors
- Posting script code on the forum and asking others to view it and suggest how to fix it
- Sleeping on it

*Don't wait till the last minute to start your project!*



# Final Project forum tips



## Use the forum effectively to get scripting help

*Not so good ...*

**Preview:**

Help!

My script is getting weird error

- Homer

*Not enough information has been provided  
on this post for others to help*

## Use the forum effectively to get scripting help

*Better ... but requires viewer to log into Opus and you may have modified the script since posting*

### Preview:

Help!

My script is getting weird error

My script is here:

/home/cis90/milhom/bin/myscript

And this is the error:

CODE: SELECT ALL

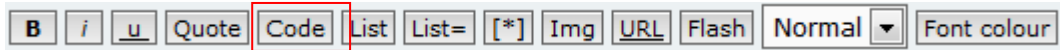
```
/home/cis90/simben/bin $ ./script99
simben90
-rwxr-x--- 1 simben90 cis90 10489 Apr 30 07:33 /home/cis90/simben/bin/myscript
./script99: line 8: unexpected EOF while looking for matching `"'
./script99: line 16: syntax error: unexpected end of file
/home/cis90/simben/bin $
```

- Homer

*This post provides the location of the script and the error message which enables others to help you find and fix the problem*



## Use the forum effectively to get scripting help



### Preview:

Help!

My script is getting weird error

This is the script:

CODE: SELECT ALL

```
#!/bin/bash
# Test script
#
echo $LOGNAME
dir=/home/cis90/simben
ls -l $dir/bin/myscript
if [ -f "$dir/bin/myscript" ]; then
    echo you have a myscript file in the bin directory
else
    echo there is no myscript file in your bin directory!
fi
exit
```

And this is the error:

CODE: SELECT ALL

```
/home/cis90/simben/bin $ ./script99
simben90
-rwxr-x--- 1 simben90 cis90 10489 Apr 30 07:33 /home/cis90/simben/bin/myscript
./script99: line 8: unexpected EOF while looking for matching `"'
./script99: line 16: syntax error: unexpected end of file
/home/cis90/simben/bin $
```

- Homer

*Best ...*

*This post shows both the script and the error using code tags which enables others to help you find and fix the problem.*

*The thread will also benefit future CIS 90 students*



# Scripting Tips

## echo

# Silence is golden

*Many UNIX commands that run successfully produce no output*

```
[simben90@opus bin]$ alias details=file  
[simben90@opus bin]$ cp quiet quiet.bak  
[simben90@opus bin]$ value=002  
[simben90@opus bin]$ umask $value  
[simben90@opus bin]$ cat quiet > /dev/null  
[simben90@opus bin]$ > important_file
```

# Silence is golden

*Running or sourcing a script full of UNIX commands that produce no output .... still produces no output!*

```
[simben90@opus bin]$ cat quiet
alias details=file
cp quiet quiet.bak
value=002
umask $value
cat quiet > /dev/null
> important_file
```

```
[simben90@opus bin]$ quiet
[simben90@opus bin]$
```

```
[simben90@opus bin]$ source quiet
[simben90@opus bin]$
```

# Silence is golden

*You can use the echo command in your scripts to provide:*

- *interaction*
- *feedback*
- *tracing (for debugging)*

```
[simben90@opus bin]$ cat quiet
alias details=file
cp quiet quiet.bak
value=002
umask $value
cat quiet > /dev/null
> important_file
```

```
[simben90@opus bin]$ quiet
[simben90@opus bin]$
```

```
[simben90@opus bin]$ cat not-so-quiet
alias details=file
cp quiet quiet.bak
value=002
umask $value
echo TRACE: value=$value
cat quiet > /dev/null
echo "Quiet script successfully completed"
```

```
[simben90@opus bin]$ not-so-quiet
TRACE: value=002
Quiet script successfully completed
```



# Scripting Tips

`$(some-command)`

## Utilizing `$(some-command)`

The **\$** metacharacter provides the "value" of both variables, e.g. `$PS1` or commands, e.g. `$(some-command)`:

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

```
/home/cis90/simben $ echo $(grep love poems/Shakespeare/* | wc -l)
11
```

```
/home/cis90/simben $ myname=$(grep $LOGNAME /etc/passwd | cut -f5 -d":")
/home/cis90/simben $ echo My name is $myname
My name is Benji Simms
```

*This is useful when you want to insert the output of a command into a sentence being echoed*



# Scripting Tips

date



# Utilizing the date command

```
/home/cis90/simben $ date  
Wed Nov 26 15:35:53 PST 2008
```

```
/home/cis90/simben $ date +%r  
04:14:26 PM
```

```
/home/cis90/simben $ time=$(date +%r)
```

```
/home/cis90/simben $ echo "At the tone the time will be $time"
```

```
At the tone the time will be 04:15:02 PM
```

```
/home/cis90/simben $ date +%A
```

```
Tuesday
```

```
/home/cis90/simben $ day=$(date +%A)
```

```
/home/cis90/simben $ echo "Today is $day"
```

```
Today is Tuesday
```

*See the man page on date for lots of other % codes*

## Class Activity

Your turn, make a script by adding the following two lines to a file named *mydate* using the vi editor:

```
echo "Hola $LOGNAME"  
echo Today is $(date +%m/%d/%Y')
```

Give the script execute permissions and run it:

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

*Copy and paste the output of your script into the chat window*



# tips on script names

# Don't name your scripts "script"

```
[simben90@opus bin]$ ls -l script  
-rwxr-x--- 1 simben90 cis90 47 Nov 23 16:44 script
```

```
[simben90@opus bin]$ cat script  
echo "Hello from the script file named script"
```

*What would happen if you ran the script above?*

# Don't name your scripts "script"

```
[simben90@opus bin]$ cat script  
echo "Hello from the script file named script"
```

```
[simben90@opus bin]$ script  
Script started, file is typescript
```



*Why the heck doesn't  
my script do what it's  
supposed to do?*

# Don't name your scripts "script"

```
[simben90@opus bin]$ cat script
echo "Hello from the script file named script"
```

```
[simben90@opus bin]$ script
Script started, file is typescript
```



*Why the heck doesn't my script do what it's supposed to do?*

```
[simben90@opus bin]$ Where is my script?
bash: Where: command not found
```

```
[simben90@opus bin]$ exit
Script done, file is typescript
```



```
[simben90@opus bin]$ cat typescript
Script started on Wed 13 May 2009 08:00:02 AM PDT
```

```
[simben90@opus bin]$ Where is my script?
bash: Where: command not found
```

```
[simben90@opus bin]$ exit
```

```
Script done on Wed 13 May 2009 08:00:47 AM PDT
```

```
[simben90@opus bin]$
```

# Don't name your scripts "script"

*Why doesn't script do what it is supposed to do? ... because script is the name of an existing UNIX command!*

```
[simben90@opus bin]$ man script
[simben90@opus bin]$
```

The screenshot shows a terminal window titled "roddyduk@opus:~/bin" displaying the manual page for the "script" command. The window title bar includes standard Linux window controls (minimize, maximize, close). The terminal content is as follows:

```
SCRIPT (1) BSD General Commands Manual SCRIPT (1)
NAME
    script - make typescript of terminal session
SYNOPSIS
    script [-a] [-c COMMAND] [-f] [-q] [-t] [file]
DESCRIPTION
    Script makes a typescript of everything printed on your terminal. It is
    useful for students who need a hardcopy record of an interactive session
    as proof of an assignment, as the typescript file can be printed out
    later with lpr(1).

    If the argument file is given, script saves all dialogue in file. If no
    file name is given, the typescript is saved in the file typescript.

Options:
    -a      Append the output to file or typescript, retaining the prior con-
           tents.
    -c COMMAND
           Run the COMMAND rather than an interactive shell. This makes it
           easy for a script to capture the output of a program that behaves
           differently when its stdout is not a tty.
```

# Don't name your scripts "script"

*There are (at least) two files named script on Opus*

```
[simben90@opus bin]$ type script
script is hashed (/usr/bin/script)
[simben90@opus bin]$ file /usr/bin/script
/usr/bin/script: ELF 32-bit LSB executable, Intel 80386, version 1
(SYSV), for GNU/Linux 2.6.9, dynamically linked (uses shared libs),
for GNU/Linux 2.6.9, stripped
```

```
[simben90@opus bin]$ type /home/cis90/simben/bin/script
/home/cis90/simben/bin/script is /home/cis90/simben/bin/script
[simben90@opus bin]$ file /home/cis90/simben/bin/script
/home/cis90/simben/bin/script: ASCII text
[simben90@opus bin]$
```

**Question:** *Why did bash run the script in /usr/bin instead of the script in /home/cis90/simben/bin?*



# Don't name your scripts "script"

**Question:** Why did bash run the script in /usr/bin instead of the script in /home/cis90/simben/bin?

The Linux **script** command is in this directory

```
[simben90@opus bin]$ echo $PATH  
/usr/kerberos/bin:/usr/local/bin:/bin:/usr/bin:/home/cis90/bin:  
/home/cis90/simben/bin:.
```

Our script, named **script**, is in this directory

**Answer:** bash searches the path in the order the directories are listed. It finds the script command in /user/bin first.

# Don't name your scripts "script"

*To override the PATH you can always specify an absolute pathname to the file you want to run:*

```
[simben90@opus bin]$ /home/cis90/simben/bin/script  
Hello from the script file named script
```

```
[simben90@opus bin]$ ./script  
Hello from the script file named script
```

*Note the shell treats the . above as "here" which in this case is /home/cis90/simben/bin*

## Try the script command

- Use the **script** command to start recording
- Type various commands of your choice
- Type **exit** or hit **Ctrl-D** to end recording
- Use **cat typescript** to see what you recorded

*This would be a good way to record a session such as working one of the lab assignments for future reference.*



# Review

```
function runningScript ()  
{
```

## The rules of the road for variables

- Rule 1: A child process can only see variables the parent has exported.
- Rule 2: A child process cannot change the parent's variables.

## Running a Script

```
/home/cis90/simben $ cat mydate  
#!/bin/bash  
echo "Hola $LOGNAME"  
date +%m/%d/%Y  
echo $myvar1 $myvar2 $myvar3
```

*Add this line to  
the last script we  
made*

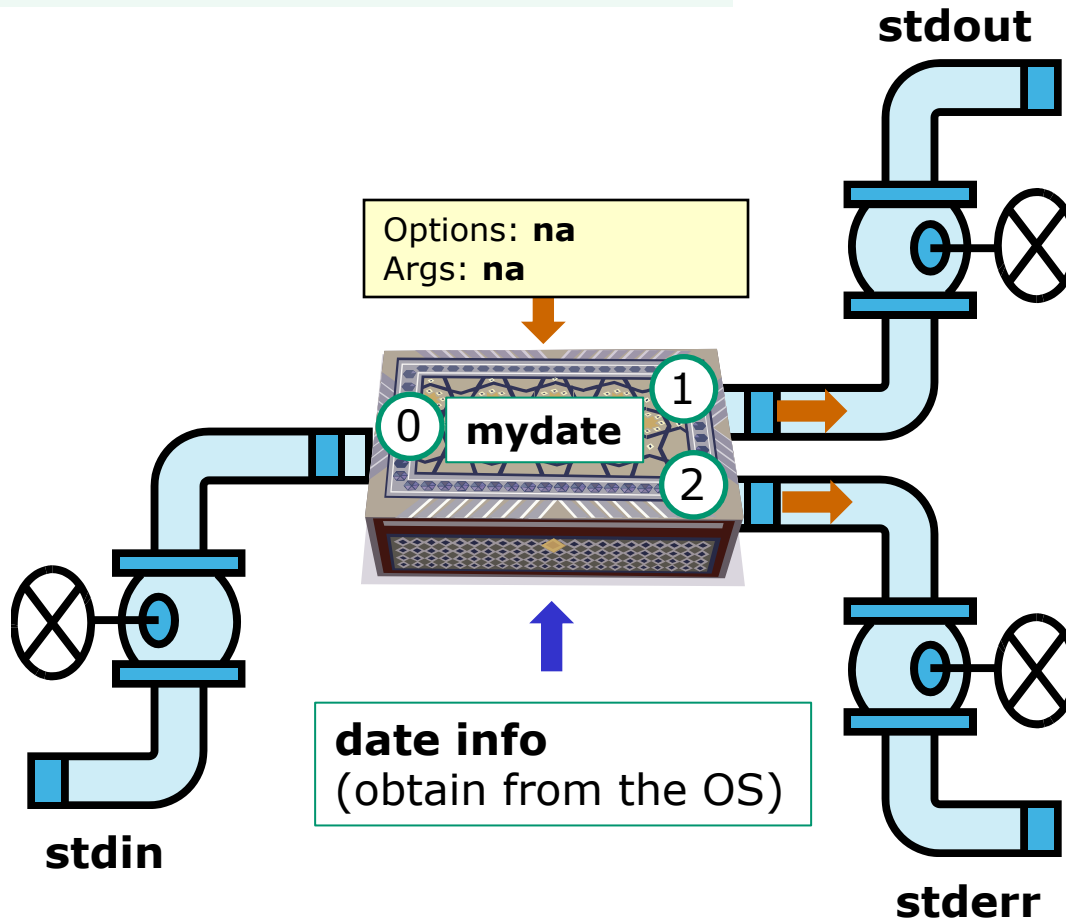
*Don't initialize  
them yet*

```
/home/cis90/simben $ mydate  
Hola simben90  
05/16/2013  
  
/home/cis90/simben $
```

*Because the variables  
don't exist yet the last  
echo statement prints a  
blank line*

# Running a Script

```
$ mydate
```



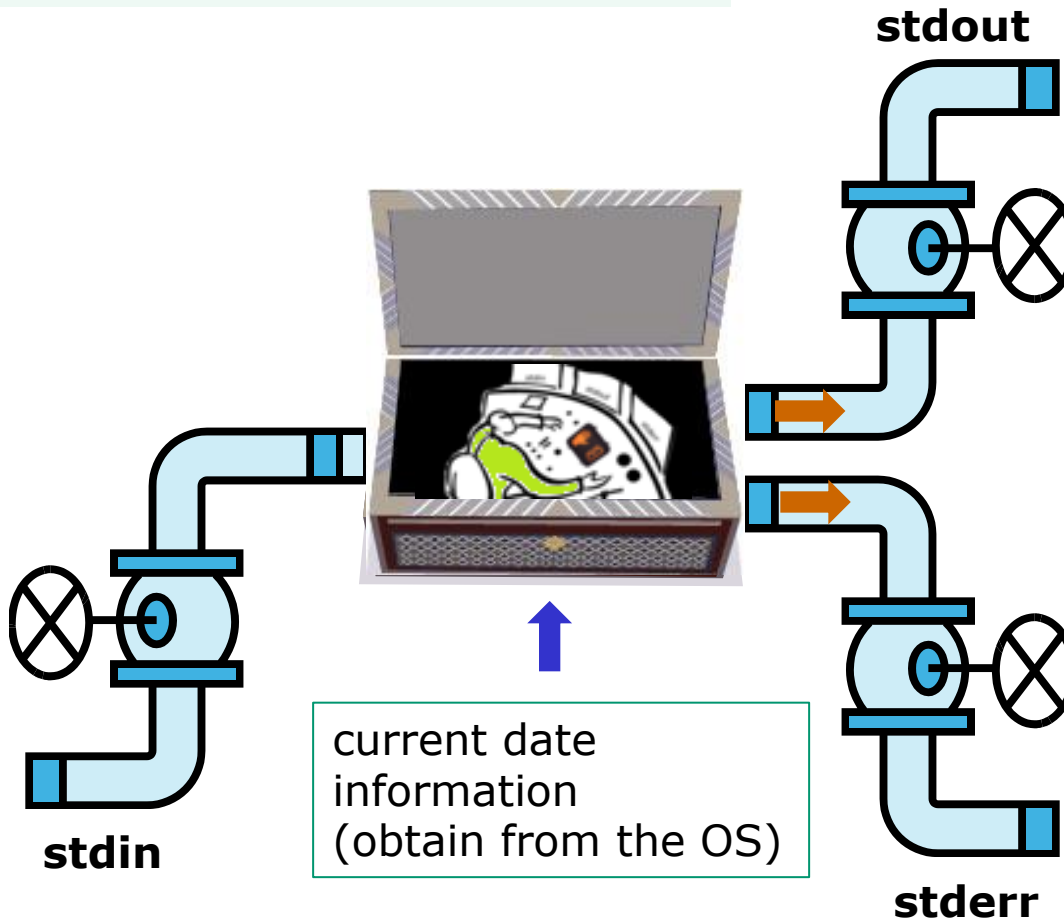
```
Hola simben90
05/09/2013
```

*In this example, output from **myscript** goes to **stdout**.*

*stdout has not been redirected so it goes to the default terminal device (your screen).*

# Running a Script

```
$ mydate
```



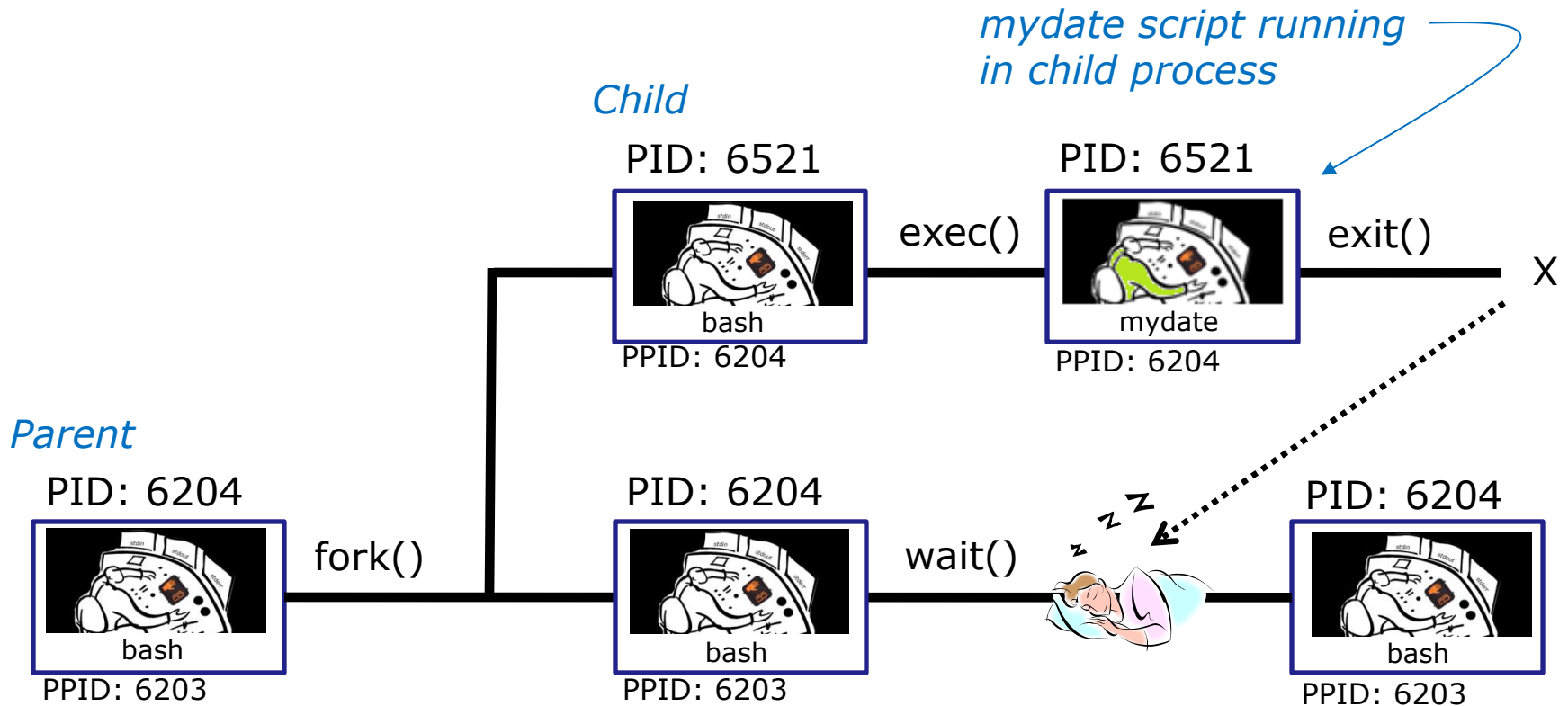
```
Hola simben90  
05/16/2012
```

*A sneak peek into memory  
to see what our process  
looks like!*





# Running a Script



Whenever you run any command, program, or script it runs as a **child process**

## Running a Script

```
/home/cis90/simben $ cat mydate  
#!/bin/bash  
echo "Hola $LOGNAME"  
date +%m/%d/%Y'  
echo $myvar1 $myvar2 $myvar3
```

*In the parent process, initialize the three variables*

```
/home/cis90/simben $ myvar1=Tic; myvar2=Tac; myvar3=Toe  
/home/cis90/simben $ echo $myvar1 $myvar2 $myvar3  
Tic Tac Toe
```

*What happens if we run **mydate** now?*

## Running a Script

```
/home/cis90/simben $ cat mydate  
#!/bin/bash  
echo "Hola $LOGNAME"  
date +%m/%d/%Y'  
echo $myvar1 $myvar2 $myvar3
```

```
/home/cis90/simben $ myvar1=Tic; myvar2=Tac; myvar3=Toe  
/home/cis90/simben $ echo $myvar1 $myvar2 $myvar3  
Tic Tac Toe
```

```
/home/cis90/simben $ mydate  
Hola simben90  
05/09/2012
```

*Running **mydate**  
(as a child process)*

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

***Why no Tic Tac Toe output?***

## Running a Script

```
/home/cis90/simben $ export myvar1  
/home/cis90/simben $ mydate  
Hola simben90  
05/09/2012  
Tic
```

*Rule 1: A child process can only see variables the parent has exported*

```
/home/cis90/simben $ export myvar2  
/home/cis90/simben $ mydate  
Hola simben90  
05/09/2012  
Tic Tac
```

```
/home/cis90/simben $ export myvar3  
/home/cis90/simben $ mydate  
Hola simben90  
05/09/2012  
Tic Tac Toe
```

## Running a Script

```
/home/cis90/simben $ echo $myvar1 $myvar2 $myvar3  
Tic Tac Toe
```

```
/home/cis90/simben $ cat mydate
```

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

```
echo "Hola $LOGNAME"
```

```
date +%m/%d/%Y'
```

```
echo $myvar1 $myvar2 $myvar3
```

```
myvar1=red myvar2=white myvar3=blue
```

```
echo $myvar1 $myvar2 $myvar3
```

*Add these  
new lines*

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

```
Hola simben90
```

```
05/09/2012
```

```
Tic Tac Toe
```

```
red white blue
```

*Rule 2: A child process  
cannot change the  
parent's variables.*

```
/home/cis90/simben $ echo $myvar1 $myvar2 $myvar3
```

```
Tic Tac Toe
```

## Running a Script

*Unless we want them to*

```
/home/cis90/simben $ echo $myvar1 $myvar2 $myvar3  
Tic Tac Toe
```

```
/home/cis90/simben $ source mydate  
Hola simben90  
05/09/2012  
Tic Tac Toe  
red white blue
```

*Sourcing a script causes the instructions to be run in the parent process. A child process is not created*

```
/home/cis90/simben $ echo $myvar1 $myvar2 $myvar3  
red white blue
```

```
}  
while no-comprende  
do  
    runningScript  
done
```



# Printers

Sneak Peak for CIS 90 Students





Two predominate types of printers

- Thermal inkjet technology
- Laser, drum, toner technology



So many ways to hook them up ...

Now:

- Network
- USB
- Wireless (Bluetooth, IR)



Back then:

- Serial cable
- Parallel printer cable



# Printer Configuration

# CUPS

Example printer configuration



Printer: HP LaserJet 1320n  
Connection: LAN

# CUPS



*The LaserJets have a web-based management utility*

A screenshot of a web browser displaying the HP LaserJet 1320 series web-based management utility. The browser address bar shows the IP address 172.30.1.14. The page has a blue header with the HP logo and the text "hp LaserJet 1320 series". Below the header, there are tabs for "Information", "Settings", and "Networking". The "Information" tab is selected, showing a "Device Status" section with a "Status: Ready" indicator and buttons for "Refresh Status", "Enter", and "Cancel Job". Below this is a "Supplies" section showing "Toner: (% Remaining)" and a progress bar for the "Black Cartridge" at 97%. At the bottom, there is a "Product Information" section with a table of device details.

| Product Information    |                         |
|------------------------|-------------------------|
| Product Name:          | hp LaserJet 1320 series |
| Formatter Number:      | JH03T2Z                 |
| Product Serial Number: | CNHC6360LV              |
| Service ID:            | 16101                   |
| Firmware Datecode:     | 20041024                |
| Total Memory:          | 16 MBytes               |

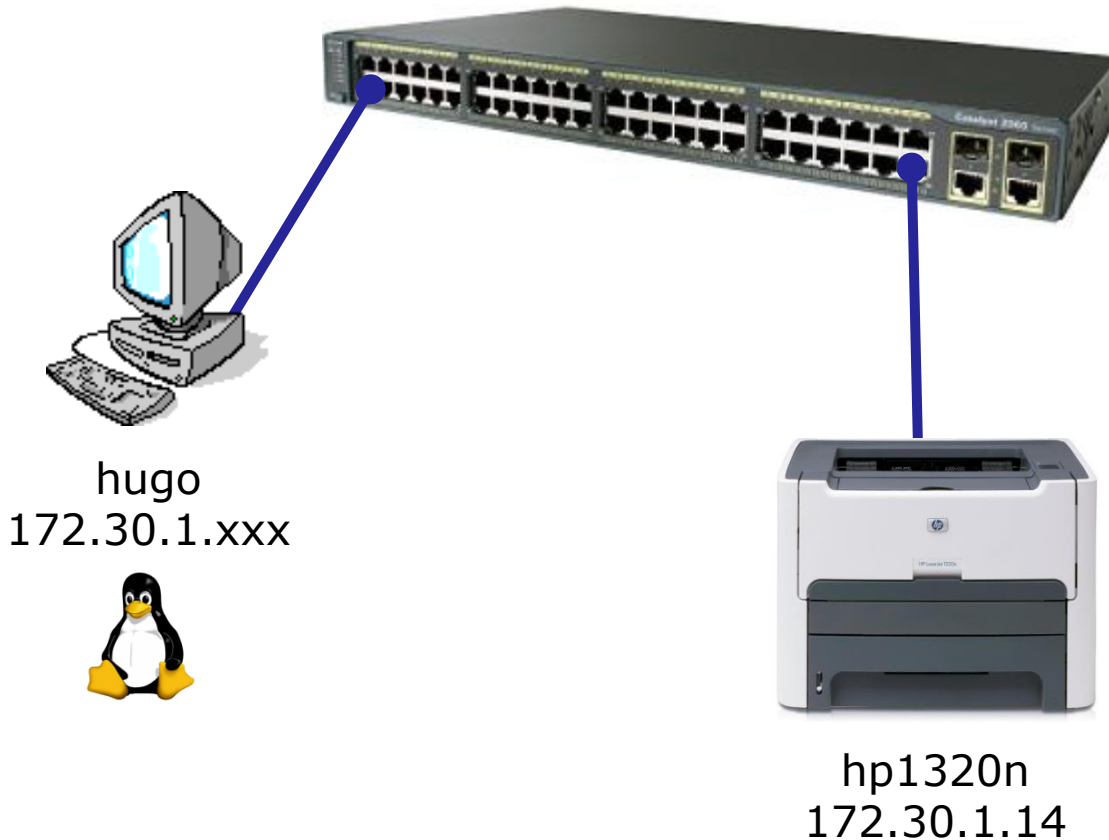
*IP Address for this 1320n  
is 172.30.1.14*



# CUPS

# CUPS

*This example will show how to add the HP 1320n as a networked printer.*



# CUPS



The image shows a terminal window titled "Hugo [Running] - Oracle VM VirtualBox". The terminal output shows the following commands and results:

```
rsimms@hugo:~$ ps -l
F S  UID  PID  PPID  C  PRI  NI ADDR  SZ  WCHAN  TTY          TIME CMD
0 S  1000  1797  1787  2  80   0  -  1777 wait   pts/0    00:00:00 bash
0 R  1000  1856  1797  0  80   0  -  1172 -      pts/0    00:00:00 ps
rsimms@hugo:~$ ps -ef | grep cups
root      674    1  0  20:24 ?        00:00:00 /usr/sbin/cupsd -F
rsimms   1878  1797  0  20:26 pts/0    00:00:00 grep --color=auto cups
rsimms@hugo:~$ firefox localhost:631 &
```

A white box with a black border is overlaid on the terminal, containing the following text:

*Access the CUPS service using a web browser with*

```
rsimms@hugo:~$ firefox localhost:631 &
```



Hugo [Running] - Oracle VM VirtualBox

Machine View Devices Help

File Edit View History Bookmarks Tools Help

Home - CUPS 1.5.2


localhost:631

Google

Home Administration Classes Online Help Jobs Printers Search Help

## CUPS 1.5.2

CUPS is the standards-based, open source printing system developed by [Apple Inc.](#) for Mac OS® X and other UNIX®-like operating systems.



### CUPS for Users

- [Overview of CUPS](#)
- [Command-Line Printing and Options](#)
- [What's New in CUPS 1.5](#)
- [User Forum](#)

### CUPS for Administrators

- [Adding Printers and Classes](#)
- [Managing Operation Policies](#)
- [Printer Accounting Basics](#)
- [Server Security](#)
- [Using Kerberos Authentication](#)
- [Using Network Printers](#)
- [cupsd.conf Reference](#)
- [Find Printer Drivers](#)

### CUPS for Developers

- [Introduction to CUPS Programming](#)
- [CUPS API](#)
- [Filter and Backend Programming](#)
- [HTTP and IPP APIs](#)
- [PPD API](#)
- [Raster API](#)
- [PPD Compiler Driver Information File Reference](#)
- [Developer Forum](#)

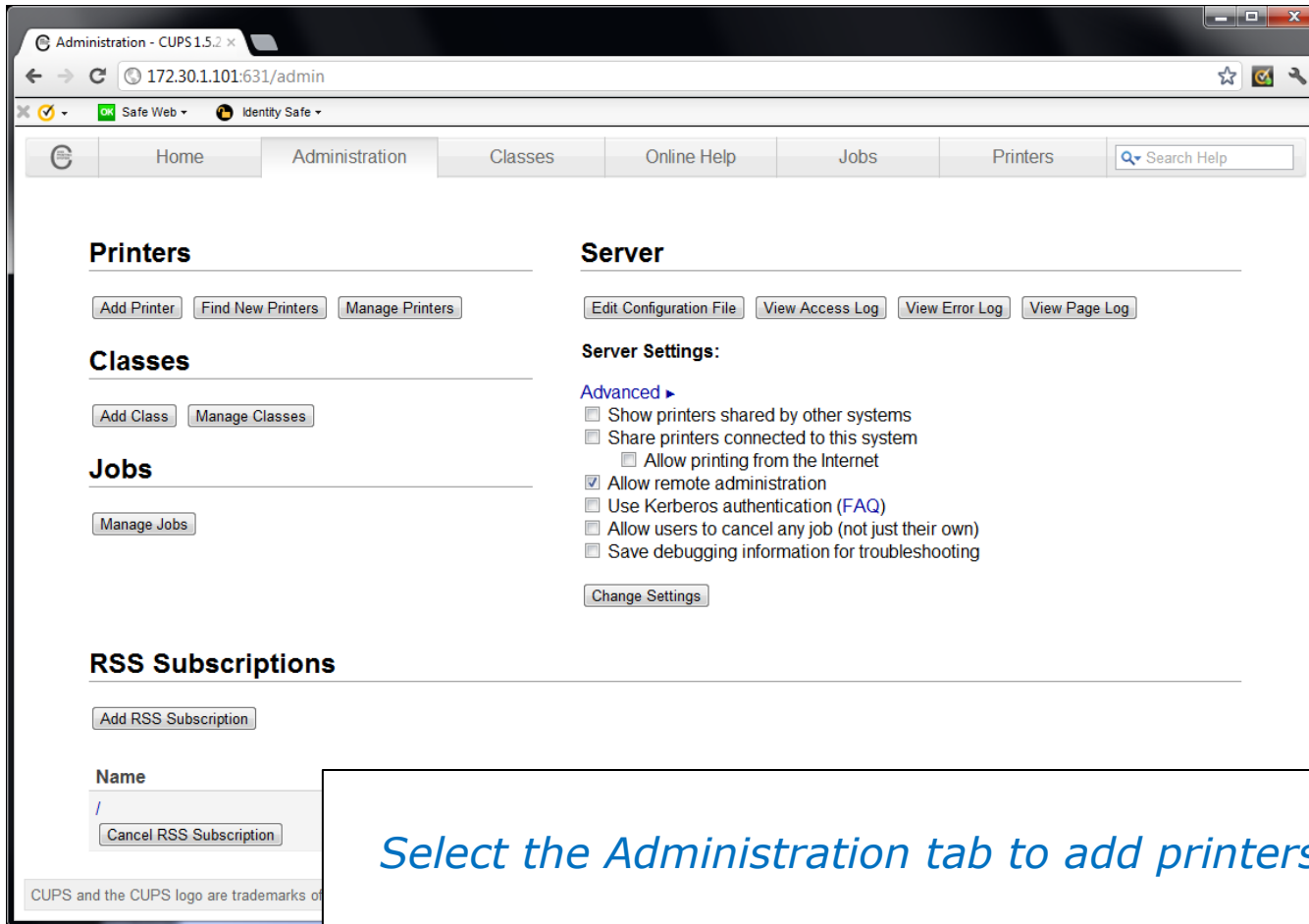
CUPS and the CUPS logo are trademarks of [Apple Inc.](#) CUPS is copyright 2007-2012 [Apple Inc.](#) All rights reserved.





The screenshot shows a web browser window with the address bar displaying "172.30.1.101:631". The page title is "Home - CUPS 1.5.2". The navigation menu includes "Home", "Administration", "Classes", "Online Help", "Jobs", and "Printers", along with a "Search Help" field. The main content area features the heading "CUPS 1.5.2" and a description: "CUPS is the standards-based, open source printing system developed by Apple Inc. for Mac OS® X and other UNIX®-like operating systems." To the right is the "UNIX PRINTING SYSTEM" logo. Below this, there are three columns of links: "CUPS for Users" (Overview of CUPS, Command-Line Printing and Options, What's New in CUPS 1.5, User Forum), "CUPS for Administrators" (Adding Printers and Classes, Managing Operation Policies, Printer Accounting Basics, Server Security, Using Kerberos Authentication, Using Network Printers, cupsd.conf Reference, Find Printer Drivers), and "CUPS for Developers" (Introduction to CUPS Programming, CUPS API, Filter and Backend Programming, HTTP and IPP APIs, PPD API, Raster API, PPD Compiler Driver Information File Reference, Developer Forum). A footer note states "CUPS and the CUPS logo are trademarks of Apple Inc., registered in the U.S. and other countries." A text box at the bottom of the screenshot contains the instruction: "Access the CUPS service remotely using a web browser on a different system".

*Access the CUPS service remotely using a web browser on a different system*



The screenshot shows a web browser window titled "Administration - CUPS 1.5.2" with the URL "172.30.1.101:631/admin". The browser has tabs for "Safe Web" and "Identity Safe". The navigation menu includes "Home", "Administration", "Classes", "Online Help", "Jobs", and "Printers", along with a "Search Help" box. The main content area is divided into several sections:

- Printers:** Contains buttons for "Add Printer", "Find New Printers", and "Manage Printers".
- Classes:** Contains buttons for "Add Class" and "Manage Classes".
- Jobs:** Contains a button for "Manage Jobs".
- RSS Subscriptions:** Contains a button for "Add RSS Subscription". Below it is a form with a "Name" field containing a slash "/" and a "Cancel RSS Subscription" button.
- Server:** Contains buttons for "Edit Configuration File", "View Access Log", "View Error Log", and "View Page Log".
- Server Settings:** Includes a section for "Advanced" settings with the following options:
  - Show printers shared by other systems
  - Share printers connected to this system
    - Allow printing from the Internet
  - Allow remote administration
  - Use Kerberos authentication (FAQ)
  - Allow users to cancel any job (not just their own)
  - Save debugging information for troubleshootingA "Change Settings" button is located below these options.

At the bottom left, a small note states: "CUPS and the CUPS logo are trademarks of..."

Select the Administration tab to add printers



The screenshot shows the CUPS 1.5.2 administration web interface. The browser address bar shows <https://172.30.1.101:631/admin/>. The interface has a navigation menu with "Home", "Administration", "Classes", "Online Help", "Jobs", and "Printers". The "Administration" section is active, showing "Printers" and "Server" tabs. Under "Printers", there are buttons for "Add Printer", "Find New Printers", and "Manage Printers". Under "Server", there are buttons for "Edit Configuration File", "View Access Log", "View Error Log", and "View Page Log". A modal dialog box titled "Authentication Required" is open in the center. It contains the text: "The server 172.30.1.101:631 requires a username and password. The server says: CUPS." Below this text are two input fields: "User Name:" with the value "rsimms" and "Password:" with a masked value "\*\*\*\*\*". At the bottom of the dialog are "Log In" and "Cancel" buttons.

*Must authenticate to add new printer*



The screenshot shows a web browser window titled "Add Printer - CUPS 1.5.2" with the URL "https://172.30.1.101:631/admin/". The browser's address bar shows "https://172.30.1.101:631/admin/". The page has a navigation menu with "Home", "Administration", "Classes", "Online Help", "Jobs", and "Printers". A search box labeled "Search Help" is also present. The main content area is titled "Add Printer" and lists three categories of printers:

- Local Printers:**
  - HP Printer (HPLIP)
  - HP Fax (HPLIP)
- Discovered Network Printers:**
  - hp LaserJet 1320 series (9C595F) (hp hp LaserJet 1320 series)
  - hp LaserJet 1320 series (9C595F) (hp hp LaserJet 1320 series)
- Other Network Printers:**
  - Backend Error Handler
  - LPD/LPR Host or Printer
  - Internet Printing Protocol (https)
  - Internet Printing Protocol (ipp)
  - Internet Printing Protocol (ipp)
  - AppSocket/HP JetDirect
  - Internet Printing Protocol (http)
  - Windows Printer via SAMBA

A "Continue" button is located at the bottom of the list.

*Nice! CUPS service already discovered a printer on the network*



The screenshot shows a web browser window titled "Add Printer - CUPS 1.5.2" with the URL <https://172.30.1.101:631/admin>. The browser's address bar shows a security warning for the URL. The page has a navigation menu with "Home", "Administration", "Classes", "Online Help", "Jobs", and "Printers", along with a "Search Help" field. The main content area is titled "Add Printer" and contains the following form fields:

- Name:**   
(May contain any printable characters except "/", "#", and space)
- Description:**   
(Human-readable description such as "HP LaserJet with Duplexer")
- Location:**   
(Human-readable location such as "Lab 1")
- Connection:** socket://172.30.1.14
- Sharing:**  Share This Printer

A "Continue" button is located below the sharing options.

*Customize printer description*



The screenshot shows a web browser window titled "Add Printer - CUPS1.5.2" with the URL <https://172.30.1.101:631/admin>. The browser's address bar shows a security warning for "Safe Web" and "Identity Safe". The page has a navigation menu with "Home", "Administration", "Classes", "Online Help", "Jobs", and "Printers", along with a "Search Help" field. The main content area is titled "Add Printer" and contains the following information:

- Name:** HP\_LaserJet\_1320\_series
- Description:** HP LaserJet 1320 series
- Location:** Family room
- Connection:** socket://172.30.1.14
- Sharing:** Do Not Share This Printer
- Make:** HP (with a dropdown menu showing "Select Another Make/Manufacturer")
- Model:** A list box containing several printer models, with "HP LaserJet 1320 Series hpjps pcl3, 3.12.2 (en)" selected.

Below the model list, there is a section "Or Provide a PPD File:" with a "Choose File" button (showing "No file chosen") and an "Add Printer" button.

*Select the printer driver*



Set Printer Options - CUPS | x

https://172.30.1.101:631/admin

Home Administration Classes Online Help Jobs Printers Search Help

### Set Default Options for HP\_LaserJet\_1320\_series

Query Printer for Default Options

**General** Printout Mode Banners Policies

**General**

Media Size: Letter

Printout Mode: Normal

Media Source: Printer default

Double-Sided Printing: Off

Set Default Options

*Set default printing options for new printer*





The screenshot shows a web browser window with the URL `https://172.30.1.101:631/printers/HP_LaserJet_1320_series`. The page has a navigation menu with tabs for Home, Administration, Classes, Online Help, Jobs, and Printers. The main content area is titled "HP\_LaserJet\_1320\_series (Idle, Accepting Jobs, Not Shared)". It includes dropdown menus for Maintenance and Administration, and the following details:

- Description:** HP LaserJet 1320 series
- Location:** Family room
- Driver:** HP LaserJet 1320 Series hpijs pcl3, 3.12.2 (color, 2-sided printing)
- Connection:** socket://172.30.1.14
- Defaults:** job-sheets=none, none media=na\_letter\_8.5x11in sides=one-sided

Below this is a "Jobs" section with a search input field labeled "Search in HP\_LaserJet\_1320\_series:" and "Search" and "Clear" buttons. At the bottom of the jobs section are buttons for "Show Completed Jobs" and "Show All Jobs". The text "No jobs." is displayed in the center.

*Ready to roll!*



The screenshot shows a web browser window with the URL `https://172.30.1.101:631/printers/HP_LaserJet_1320_series`. The browser interface includes a navigation menu with options: Home, Administration, Classes, Online Help, Jobs, and Printers. A search bar is present next to the Printers menu item.

### HP\_LaserJet\_1320\_series (Processing, Accepting Jobs, Not Shared)

Maintenance: [dropdown] Administration: [dropdown]

**Description:** HP LaserJet 1320 series  
**Location:** Family room  
**Driver:** HP LaserJet 1320 Series hpijs pcl3, 3.12.2 (color, 2-sided printing)  
**Connection:** socket://172.30.1.14  
**Defaults:** job-sheets=none, none media=na\_letter\_8.5x11in sides=one-sided

#### Jobs

Search in HP\_LaserJet\_1320\_series: [input] [Search] [Clear]

[Show Completed Jobs] [Show All Jobs]

Showing 1 of 1 active job.

| ID                        | Name    | User     | Size | Pages   | State            | Control                 |
|---------------------------|---------|----------|------|---------|------------------|-------------------------|
| HP_LaserJet_1320_series-1 | Unknown | Withheld | 1k   | Unknown | processing since | [Cancel Job] [Move Job] |

*Printing a test page*



The screenshot shows a web browser window with the address bar displaying `https://172.30.1.101:631/printers/HP_LaserJet_1320_series`. The page title is "HP\_LaserJet\_1320\_series (Idle, Accepting Jobs, Not Shared)". Below the title, there are two dropdown menus for "Maintenance" and "Administration". The main content area lists the following details:

- Description:** HP LaserJet 1320 series
- Location:** Family room
- Driver:** HP LaserJet 1320 Series hpijs pcl3, 3.12.2 (color, 2-sided printing)
- Connection:** socket://172.30.1.14
- Defaults:** job-sheets=none, none media=na\_letter\_8.5x11in sides=one-sided

Below this information is a "Jobs" section with a search bar labeled "Search in HP\_LaserJet\_1320\_series:" and "Search" and "Clear" buttons. At the bottom of the jobs section are two buttons: "Show Completed Jobs" and "Show All Jobs". The text "No jobs." is centered below these buttons.

*Printed ... this printer is ready to go!*



# Printing in Linux

# Printing Commands

## The ATT System V way

- lp (to print)
- lpstat (queue management)
- cancel (to remove jobs)

## The BSD (Berkeley Software Distribution) way

- lpr (to print)
- lpq (queue management)
- lprm (to remove jobs)

*BSD is a branch of UNIX that was developed at the University of California, Berkeley*

## And now CUPS ...

- Provides both System V and Berkeley based command-line interfaces
- Supports new Internet Printing Protocol
- Works with Samba

# CUPS

## lpstat command

Syntax: **lpstat** [*options*]

```
rsimms@hugo:~$ lpstat -p  
printer HP_LaserJet_1320_series is idle.  enabled since Tue 08 May  
2012 08:46:45 PM PDT
```

*The -p option will show the  
available printers*

```
rsimms@hugo:~$ lpstat -p -d  
printer HP_LaserJet_1320_series is idle.  enabled since Tue 08 May  
2012 08:46:45 PM PDT  
system default destination: HP_LaserJet_1320_series
```

*The -d option will identify  
the default printer*

# CUPS

## lpstat command

*On Opus*

What printers are available on Opus?

Which is the default printer?

*Write your answers in the chat window*

# CUPS

## lp and lpr commands

*Use **lp** (or **lpr**) to print files*

```
/home/cis90/simben $ lp lab10  
request id is hplaser-5 (1 file(s))
```

```
/home/cis90/simben $ lp -d hplaser lab10  
request id is hplaser-6 (1 file(s))
```

*With **lp**, use the **-d** option to manually select the printer*

```
/home/cis90/simben $ lpr lab10
```

```
/home/cis90/simben $ lpr -P hplaser lab10
```

*With **lpr**, use the **-P** option to manually select a printer*



# CUPS

## lp and lpr commands

```
/home/cis90/simben $ echo "Print Me Quietly" | lpr -P hplaser  
/home/cis90/simben $
```

*Note that both lp and lpr will read from stdin.*

*This allows output from another command to be piped in*

# CUPS

## Practice Printing

*On Opus, print your lab10 and letter files*

```
lp lab10  
lpstat
```

```
lpr letter  
lpstat
```

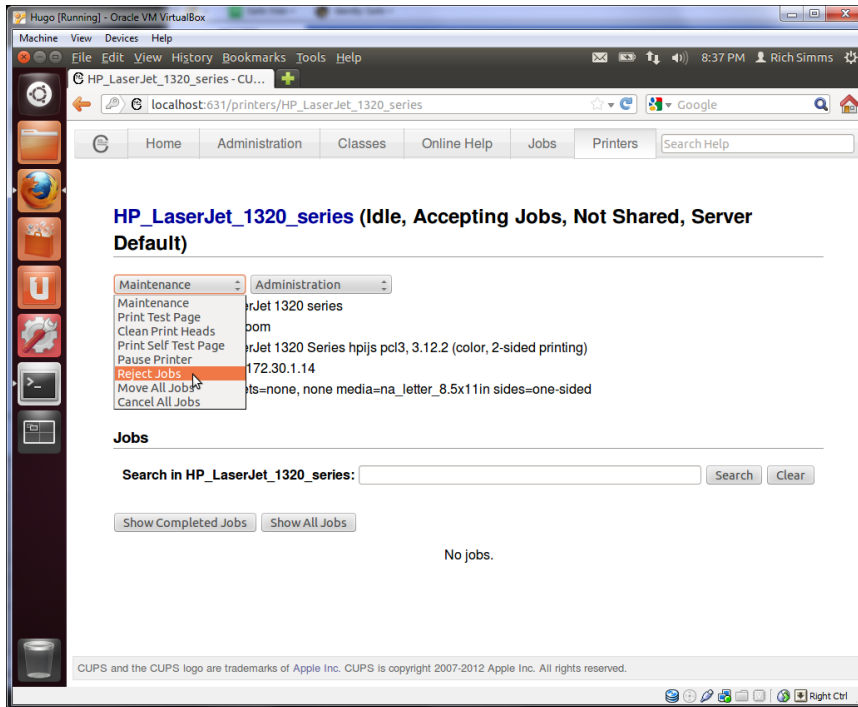
```
echo "Print Me Quietly" | lpr -P hplaser  
lpstat
```



# Managing Print Jobs

# CUPS

## Rejecting Jobs



*Clicking the **Reject Jobs** selection on the web based utility will reject further jobs*

```
[root@benji ~]# lp myfile
lp: Destination "hp7550" is not accepting jobs.
[root@benji ~]#
```

```
[root@benji ~]# lpr myfile
lpr: Destination "hp7550" is not accepting jobs.
[root@benji ~]#
```

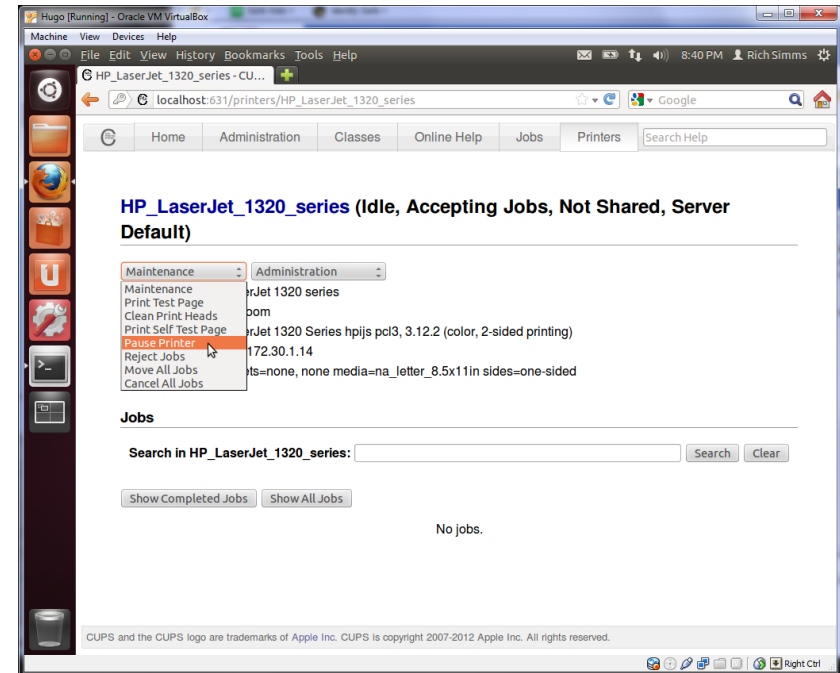
# CUPS

## Pausing the Printer

```
[root@benji ~]# lp myfile
request id is hp7550-22 (1 file(s))
```

```
[root@benji ~]# lpq
hp7550 is not ready
Rank      Owner    Job      File(s)
Total Size
1st       root    22      myfile
1024 bytes
```

```
[root@benji ~]# lpstat
hp7550-22          root
1024      Sat 15 Nov 2008 12:20:23 PM
PST
```



*Clicking the **Pause Printer** selection on the web based utility will still allow jobs to be spooled*



# CUPS

## Showing jobs waiting to print

```
[root@benji ~]# lpq
hp7550 is not ready
Rank      Owner    Job      File(s)
Total Size
1st       root    22       myfile
1024 bytes
2nd       root    23       myfile
1024 bytes
3rd       root    24       myfile
1024 bytes
4th       root    25       myfile
1024 bytes
```

*Use **lpq** or **lpstat** with no options to show spooled print jobs*

```
[root@benji ~]# lpstat
hp7550-22                root                1024    Sat
15 Nov 2008 12:20:23 PM PST
hp7550-23                root                1024    Sat
15 Nov 2008 12:20:28 PM PST
hp7550-24                root                1024    Sat
15 Nov 2008 12:20:31 PM PST
hp7550-25                root                1024    Sat
15 Nov 2008 12:20:34 PM PST
```

# CUPS

## Removing/canceling pending print jobs

```
[root@benji ~]# lpq
hp7550 is not ready
Rank   Owner   Job    File(s)
Total Size
1st    root    22     myfile
1024 bytes
2nd    root    23     myfile
1024 bytes
3rd    root    24     myfile
1024 bytes
4th    root    25     myfile
1024 bytes
```

```
[root@benji ~]# cancel 22
[root@benji ~]# cancel 23
[root@benji ~]# lprm 24
[root@benji ~]# lprm 25
```

*Use **cancel** or **lprm**  
to remove print jobs*

```
[root@benji ~]# lpq
hp7550 is not ready
no entries
```

```
[root@benji ~]# lpstat
[root@benji ~]#
```

# CUPS

## Practice Printing

### *On Opus*

```
lpq  
lpstat
```

```
cancel <print job number>  
lpq
```

```
lprm <print job number>  
lpq
```





# Wrap up

Commands:

|              |                       |
|--------------|-----------------------|
| lp, lpr      | - Linux print command |
| cancel, lprm | - cancel print job    |
| lpq, lpstat  | - Show print queue    |

Web:

|                                                         |                                     |
|---------------------------------------------------------|-------------------------------------|
| <a href="http://hostname:631">http://hostname:631</a>   | - CUPS web based management utility |
| <a href="http://hostname:9100">http://hostname:9100</a> | - HP JetDirect printer              |

## Next Class

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

*No Quiz  
No Lab due*

**Work on final project - due in two weeks!**

Optional extra credit labs

## Project Workshop

- See if you can get one “starter” task scripted and working before leaving class today.
- Grade your starter script using the Final Project rubric

Implementing all five tasks (6 points each):

- Requirements for each task:
  - Minimum of 10 “original” script command lines
  - Has one or more non-generic comments to explain what it is doing
  - Has user interaction

You don't have to do all of these but do at least five:

- Redirecting stdin (5 points)
- Redirecting stdout (5 points)
- Redirecting stderr (5 points)
- Use of permissions (5 points)
- Use of filename expansion characters (5 points)
- Use of absolute path (5 points)
- Use of relative path (5 points)
- Use of a PID (5 points)
- Use of inodes (5 points)
- Use of links (5 points)
- Use of scheduling (5 points)
- Use of a GID or group (5 points)
- Use of a UID or user (5 points)
- Use of a /dev/tty device (5 points)
- Use of a signal (5 points)
- Use of piping (5 points)
- Use of an environment variable (5 points)
- Use of /bin/mail (5 points)
- Use of a conditional (5 points)

The maximum for this section is 25 points.



# Backup