



Lesson Module Checklist

- Slides –
- Properties –
- Flash cards –
- First minute quiz – NA
- Web calendar summary –
- Web book pages –
- Commands –
- Lab –

- CCC Confer wall paper –

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

- Pickup Polycom 831.479.6392 –
- Check that room headset is charged –



Dieskau



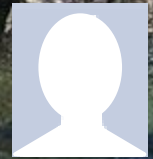
Jonathan



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



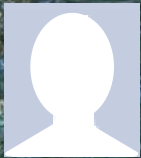
Ana



David



Obie



Dave



Cole



Corey



Nancy



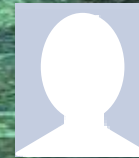
Ryan



Elia



Tasha



Darren



Scott



Devin



Everett



Juan



Raven



Rogan



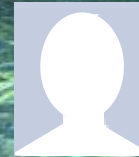
Mike



Mook



Melissa



Cameron



Jose



Jeff



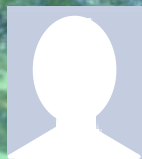
Matt



Kenneth



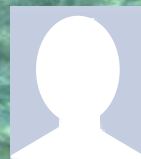
Ousmane



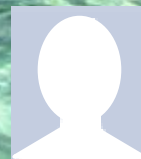
Ian



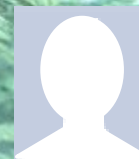
Solomon



Henry



Matthew



Mason



Chan

Quiz

**No Quiz
Today !**



- [] Has the phone bridge been added?
- [] Is recording on?
- [] Does the phone bridge have the mike?
- [] Share slides, putty x 3, Chrome and Eko VM
- [] Disable spelling on PowerPoint

More Shell Scripting

Objectives	Agenda
• Use conditionals in scripts • Transfer files between computers • Archive directories using tar	• No Quiz • Questions from last week • Getting started (if you haven't already) • Scripting tips • scp • Tarballs • Wrap up

* = hands on exercise for topic

Questions

Questions

Any questions on:

- Project?
- Extra credit Labs?
- Previous course material?



Housekeeping

Next Class

**Project is due
next week!**

Previous material and assignment

1. No labs due today
2. Project is due 11:59PM on 5/23. That's one week from now. If you haven't started yet, now would be a good time!
3. Extra credit labs are due 11:59PM 5/30.
4. Final Exam (Test #3) is 5/30 1-3:50PM in 2501 (no CCC Confer)

Current Progress			Quizzes & Tests										Forum				Labs				Project		Extra Credit	Total Grade							
Code	Name	Grading Choice	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	T1	T2	T3	F1	F2	F3	F4	L1					L2	L3	L4	L5	L6	L7	L8
	Max Points		3	3	3	3	3	3	3	3	3	3	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	60	90	560
adaldrida	WYPR	Grade	3	3	3	3	3	3	3	3	3	3	28	20	20	20	20	20	20	20	26	26	27	30	29	30	28	28	28	28	19
alatar	Grade	2	3	3	3	3	3	3	3	3	3	3	28	26	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	25
amroth	Grade	3	3	3	3	3	3	3	3	3	3	3	28	12	20	20	20	20	20	20	29	23	30	20	27	28	20	27	28	20	20
arador	Grade	3	3	3	3	3	3	3	3	3	3	3	25	21	20	20	20	20	20	19	30	24	29	30	30	30	30	30	30	29	
aragorn	Grade	3	3	3	3	3	3	3	3	3	3	3	26	30	20	20	20	20	20	28	15	26	26	30	30	30	30	30	30	61	
arwen	Grade	3	3	3	3	3	3	3	3	3	3	3	27	28	20	20	20	20	20	20	29	20	29	29	30	30	30	30	30	30	66
celebrian	Grade	2	3	3	3	3	3	3	3	3	3	3	23	27	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20
cirdan	Grade	3	3	3	3	3	3	3	3	3	3	3	29	28	20	20	20	20	20	28	29	30	24	30	30	30	30	30	30	60	
dori	Grade	3	2	2	3	3	3	3	3	3	3	3	23	23	16	20	12	28	27	30	30	30	30	30	30	29	29	29	29	76	
dwalin	Grade	3	1	1	1	1	1	1	1	1	1	1	18	22	20	20	20	20	20	20	25	20	22	18	29	20	27	28	28	25	
elrond	Grade	3	3	3	3	3	3	3	3	3	3	3	28	30	20	20	20	20	20	30	20	29	30	30	30	30	30	30	30	30	30
eomer	Grade	2	2	1	2	3	3	3	3	3	3	3	18	10	20	20	20	20	20	21	25	30	23	23	28	20	25	30	30	30	30
eowyn	Grade		2	3	3	3	3	3	3	3	3	3	26	20	20	20	20	20	20	26	28	30	15	30	30	28	20	30	30	25	25
frodo	Grade	3	3	3	3	3	3	3	3	3	3	3	26	28	20	20	20	20	20	28	24	29	30	30	30	30	30	30	30	61	
galadriel	Grade	3	3	3	3	3	3	3	3	3	3	3	22	12	20	20	20	20	20	25	20	18	24	19	23	20	20	20	20	20	20
haldir	Grade	3	0	0	3	3	3	3	3	3	3	3	26	25	20	20	20	20	20	30	20	28	20	30	30	27	20	30	30	30	30
ioeth	Grade	3	3	3	3	3	3	3	3	3	3	3	12	8	8	0	0	21	30	30	30	30	30	30	30	30	30	30	30	30	5
khamul	Grade	3	2	3	2	3	3	3	3	3	3	3	16	22	20	20	20	20	20	29	22	30	22	30	30	30	30	30	30	30	30
nessa	Grade	3	3	3	3	3	3	3	3	3	3	3	24	23	20	20	20	20	20	30	24	20	20	20	20	20	20	20	20	20	20
orome	Grade	3	3	3	3	3	3	3	3	3	3	3	27	24	20	20	20	20	20	30	27	20	25	30	30	30	30	30	30	30	30
pippin	Grade	3	3	3	3	2	3	3	3	3	3	3	27	20	20	20	20	20	20	30	20	30	30	30	30	30	30	30	30	30	15
quickbeam	Grade	3	3	3	3	3	3	3	3	3	3	3	27	29	20	20	20	20	20	25	29	30	28	29	26	30	30	30	30	23	
samwise	Grade	2	3	3	3	3	3	3	3	3	3	3	23	12	4	8	0	20	16	24	30	30	30	30	30	30	30	30	30	30	13
shadowfax	Grade	3	3	3	3	3	3	3	3	3	3	3	27	29	20	20	20	20	20	29	30	28	28	30	30	30	30	30	30	30	60
strider	Grade	3	3	3	3	3	3	3	3	3	3	3	28	27	20	20	20	20	20	30	20	20	20	30	30	30	30	30	30	30	30
theoden	Grade	1	0	3	2	3	3	3	3	3	3	3	23	20	0	8	20	27	22	29	25	27	20	30	28	28	28	28	28	28	7
treebeard	Grade	3	3	3	3	3	3	3	3	3	3	3	28	24	16	20	20	20	20	28	25	30	29	30	30	30	30	30	30	30	35
tulkas	Grade	3	2	0	3	3	3	3	3	3	3	3	22	23	0	0	20	30	22	18	29	24	27	20	28	28	28	28	28	28	40
ulmo	Grade	3	3	3	3	3	3	3	3	3	3	3	27	28	20	20	20	20	20	28	25	30	29	30	30	30	30	30	30	30	30

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

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

Tally as of 5/17/2012

adaldrida: 90% (405 of 450 points)
 alatar: 95% (430 of 450 points)
 amroth: 84% (379 of 450 points)
 arador: 92% (414 of 450 points)
 aragorn: 106% (481 of 450 points)
 arwen: 112% (507 of 450 points)
 celebrian: 73% (331 of 450 points)
 cirdan: 110% (498 of 450 points)
 dori: 102% (461 of 450 points)
 dwalin: 80% (360 of 450 points)
 elrond: 104% (472 of 450 points)
 eomer: 82% (373 of 450 points)
 eowyn: 84% (382 of 450 points)
 frodo: 110% (495 of 450 points)

gwaihir: 83% (376 of 450 points)
 haldir: 85% (383 of 450 points)
 ioreth: 20% (90 of 450 points)
 khamul: 99% (449 of 450 points)
 nessa: 82% (371 of 450 points)
 orome: 98% (442 of 450 points)
 pippin: 43% (197 of 450 points)
 quickbeam: 101% (455 of 450 points)
 samwise: 36% (166 of 450 points)
 shadowfax: 103% (465 of 450 points)
 strider: 108% (489 of 450 points)
 theoden: 68% (307 of 450 points)
 treebeard: 106% (477 of 450 points)
 tulkas: 84% (378 of 450 points)
 ulmo: 103% (466 of 450 points)

Remaining point earning opportunities = 30 + 20 + 60 = 110 points plus up to 90 points maximum extra credit

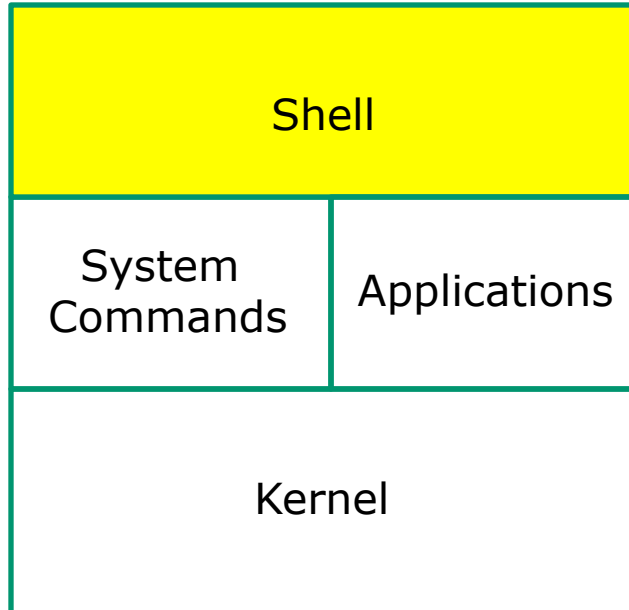
Make backup copies of your script

```
/home/cis90/simben $ cd bin  
/home/cis90/simben/bin $ cp myscript myscript.v1  
/home/cis90/simben/bin $ cp myscript myscript.v2  
/home/cis90/simben/bin $ cp myscript myscript.v3  
/home/cis90/simben/bin $
```

Refresh

UNIX/Linux Architecture

The Shell



- Allows users to interact with the computer via a "command line".
- Prompts for a command, parses the command, finds the right program and gets that program executed.
- Called a "shell" because it hides the underlying operating system.
- Many shell programs are available: sh (Bourne shell), bash (born again shell), csh (C shell), ksh (Korn shell).
- **A user interface and a programming language (scripts).**
- GNOME and KDE desktops could be called graphical shells



Shell Scripts

Some scripts on opus

- 1) `/home/cis90/bin/riddle1`
- 2) `/home/cis90/bin/allscripts`
- 3) `/etc/rc.d/init.d/network`
- 4) `/usr/bin/spell`
- 5) `/usr/bin/vimtutor`
- 6) `~/bin/enlightenment`

You have read permission for all these scripts. You can use `cat`, `more`, `less`, or even `vi` to view them

Class Exercise

Scripting

How many of the commands in /bin are really scripts?

```
file /bin/*
```

```
file /bin/* | grep script
```

```
file /bin/* | grep script | wc -l
```

How many of the commands in /usr/bin are really scripts?

```
file /usr/bin/*
```

```
file /usr/bin/* | grep script
```

```
file /usr/bin/* | grep script | wc -l
```

Project

Getting Started

Getting started on the final project

(If you haven't done this already)

1. Create a file in your bin directory named myscript:
 - Copy from /home/cis90/depot/myscript
 - or copy and paste template code from:
<http://simms-teach.com/docs/cis90/cis90final-project.pdf>
2. Give yourself full permissions and give CIS 90 group read and execute permissions
 - **chmod 750 myscript**
3. Run **allscripts** and verify your script will run without any errors

Instructor reminder: run testscripts to see current status

```
simmsben@opus:~  
/home/cis90ol/simmsben $ allscripts  
  
*****  
*           Spring 2011 CIS 90 Online Projects           *  
*****  
1) Bobby      7) Eric      13) Josh      19) Terry  
2) Chris      8) Gabriel   14) Marisol  20) Tommy  
3) Craig      9) Geoffrey   15) Merrick  21) Yu-Chen  
4) Dan M.     10) Jason     16) Quinton  
5) Daniel W.  11) Jeff      17) Tajvia  
6) Emanuel    12) Jesse     18) Tanner  
  
*****  
*           Examples and Hall of Fame           *  
*****  
50) Duke      51) Benji     52) Junior   53) Janet  
  
99) Exit  
  
Enter Your Choice: █
```

Verify that you can run
your **mymisc** from
allscripts

```
simmsben@opus:~  
  
Benji, please Enter an option number from the list below:  
  
1) What is today?  
2) The users on opus.cabrillo.edu  
3) Warning, don't go here!!  
4) Sort current directory  
5) Back pat eCards  
6) Check IP forwarding status  
  
or enter Q to Quit  
  
Enter Your Choice: █
```

Final Project

What is allscripts and myscript?

```
#!/bin/bash
#
# menu: A simple menu template
#
while true
do
    clear
    echo -n "
    Spring 2009 CIS 90 Projects
    1) Bilal
    2) Craig
    3) Dan
    4) Doug
    5) Duke
    6) Edgar D.
    7) Edgar D.
    8) Gabriel
    9) George
    10) Glen
    11) Jaime
    12) Janet
    13) Joe F.
    14) Joe F.
    15) Junious
    16) Kang
    17) Liewen
    18) Linda
    19) Michael
    20) Patrick
    21) Talley
    22) Todd
    23) William
    24) Benji
    99) Exit

    Enter Your Choice: "
    read RESPONSE
    case $RESPONSE in
        1) # Bilal
            /home/cia90/bilal/bin/myscript
            ;;
        2) # Craig
            /home/cia90/craig/bin/myscript
            ;;
        3) # Dan
            /home/cia90/dan/bin/myscript
            ;;
        4) # Doug
            /home/cia90/doug/bin/myscript
            ;;
        5) # Duke
            /home/cia90/duke/bin/myscript
            ;;
        6) # Edgar D.
            /home/cia90/edgar/bin/myscript
            ;;
        7) # Edgar D.
            /home/cia90/edgar/bin/myscript
            ;;
        8) # Gabriel
            /home/cia90/gabriel/bin/myscript
            ;;
        9) # George
            /home/cia90/george/bin/myscript
            ;;
        10) # Glen
            /home/cia90/glen/bin/myscript
            ;;
        11) # Jaime
            /home/cia90/jaime/bin/myscript
            ;;
        12) # Janet
            /home/cia90/janet/bin/myscript
            ;;
        13) # Joe F.
            /home/cia90/joe/bin/myscript
            ;;
        14) # Joe F.
            /home/cia90/joe/bin/myscript
            ;;
        15) # Junious
            /home/cia90/junious/bin/myscript
            ;;
        16) # Kang
            /home/cia90/kang/bin/myscript
            ;;
        17) # Liewen
            /home/cia90/liewen/bin/myscript
            ;;
        18) # Linda
            /home/cia90/linda/bin/myscript
            ;;
        19) # Michael
            /home/cia90/michael/bin/myscript
            ;;
        20) # Patrick
            /home/cia90/patrick/bin/myscript
            ;;
        21) # Talley
            /home/cia90/talley/bin/myscript
            ;;
        22) # Todd
            /home/cia90/todd/bin/myscript
            ;;
        23) # William
            /home/cia90/william/bin/myscript
            ;;
        24) # Benji
            /home/cia90/benji/bin/myscript
            ;;
        99) 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
```

```
#
# 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
    esac
    echo "Please enter a number between 1 and 6"
    ;;
esac
echo -n "Hit the Enter key to return to menu "
read dummy
done
```

allscripts

```
#!/bin/bash
#
# menu: A simple menu template
#
```

```
while true
do
```

```
    echo -n "
```

The while statement in allscripts will loop through the code forever

```
40) Songul
```

```
read RESPONSE
case $RESPONSE in
```

A case statement is used to run the appropriate myscript file in the student's bin directory

```
40)      # Songul
```

/home/cis90/messison/bin/myscript

```
12) # James
13) # Joe
14) # Joe P.
15) # Jonathan
16) # King
17) # Liawen
18) # Liawen
19) # Michael
20) # Michael
21) # Michael
22) # Michael
23) # Michael
24) # Michael
```

```
99)      exit 0
```

For case 99 the exit command is called which causes the script to terminate. The return code of 0 means successful

```
esac
```

```
done
```

myscript

```
#
# 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
```

The outer while statement that loops forever

myscript

```
#
# 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
```

*This is a single echo command that prints
a menu for the user*

myscript

```
#
# 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
```

This is a case statement. One case for each task.

myscript

```
#
# 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
```

read commands get input from the user

How projects are graded

Possible Points	Requirements
30	Implementing all five tasks (6 points each): - Requirements for each task: - 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: - 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: - Doesn't give full credit 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)

This is how the final project will be graded



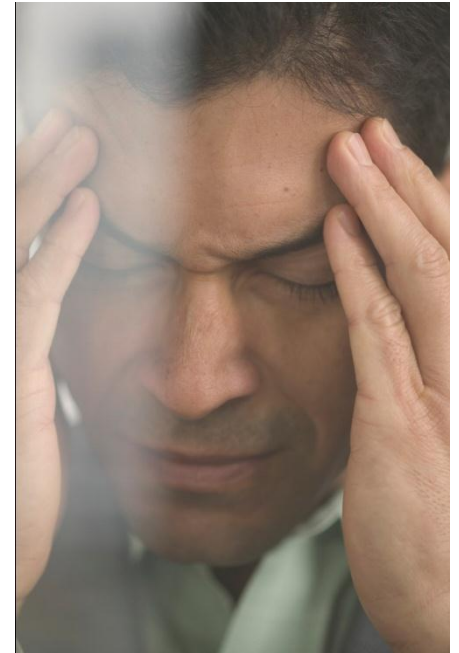
Project

Scripting a task

What takes longer?



Writing the script?
Or deciding what to script?

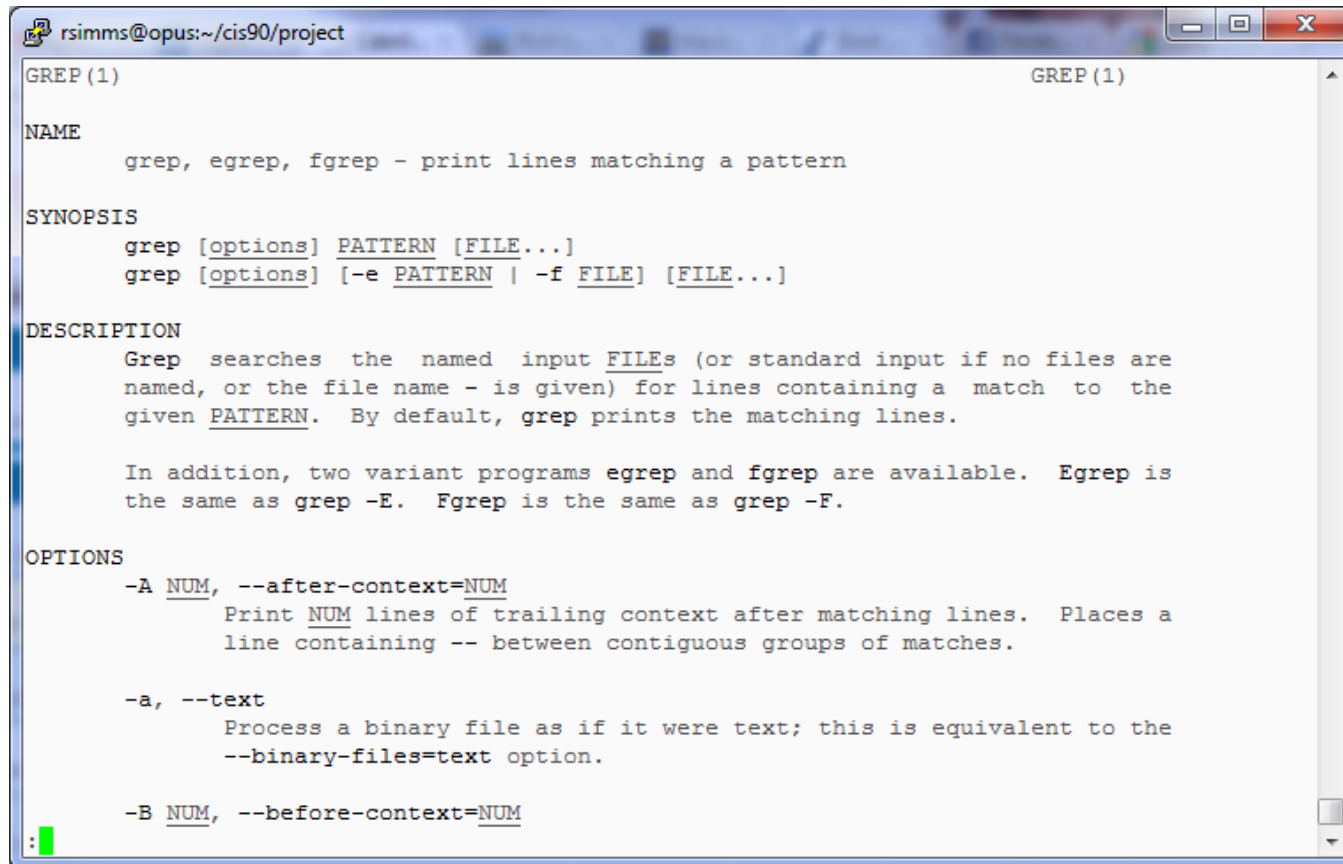


One way to get started ... select a random command to highlight with a custom script

Commands

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

This example will focus on the **grep** command.
Research your command by reading the man page



```
rsimms@opus:~/cis90/project
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 the file name - is given) 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.

OPTIONS
    -A NUM, --after-context=NUM
        Print NUM lines of trailing context after matching lines. Places a
        line containing -- between contiguous groups of matches.

    -a, --text
        Process a binary file as if it were text; this is equivalent to the
        --binary-files=text option.

    -B NUM, --before-context=NUM
```

Using **man grep** to show the man page for grep

Find some interesting options to play with

-R, -r, --recursive

Read all files under each directory, recursively; this is equivalent to the -d recurse option.

-w, --word-regexp

Select only those lines containing matches that form whole words. The test is that the matching substring must either be at the beginning of the line, or preceded by a non-word constituent character. Similarly, it must be either at the end of the line or followed by a non-word constituent character. Word-constituent characters are letters, digits, and the underscore.

-l, --files-with-matches

Suppress normal output; instead print the name of each input file from which output would normally have been printed. The scanning will stop on the first match.



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

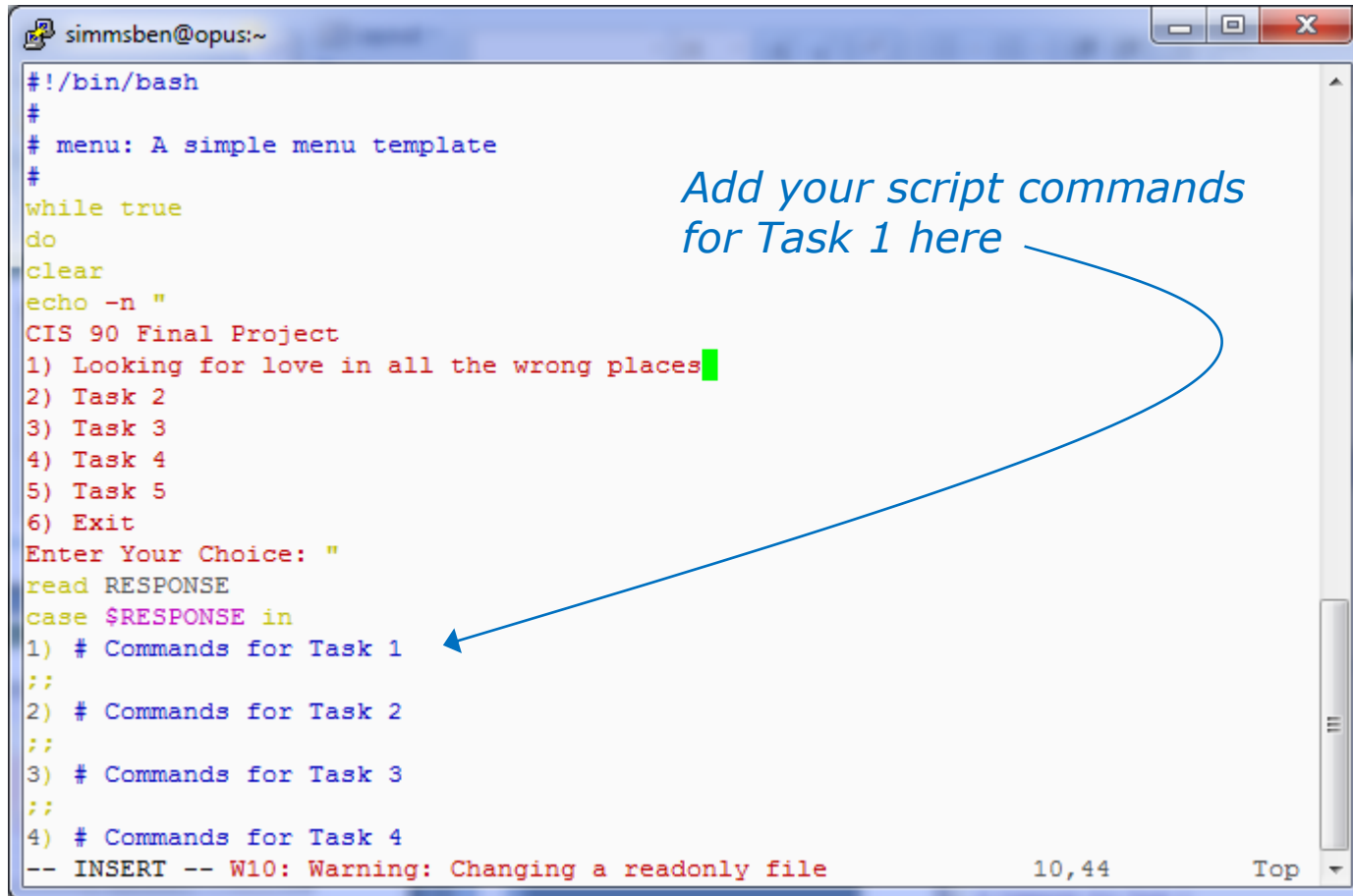
1. Write a script to call the **grep** command
2. Try the different options
3. Iterate till you are happy with it

Start hacking!

Customize the menu options for Task 1

```
simmsben@opus:~
#!/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
"/home/cis90/depot/myscript"
```

```
simmsben@opus:~
#!/bin/bash
#
# menu: A simple menu template
#
while true
do
clear
echo -n "
CIS 90 Final Project
1) Looking for love in all the wrong places
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
-- INSERT -- W10: Warning: Changing a readonly file 10,44 Top
```



The image shows a terminal window titled 'simmsben@opus:~'. The script content is as follows:

```
#!/bin/bash
#
# menu: A simple menu template
#
while true
do
clear
echo -n "
CIS 90 Final Project
1) Looking for love in all the wrong places
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
-- INSERT -- W10: Warning: Changing a readonly file
```

A blue arrow points from the text 'Add your script commands for Task 1 here' to the line '# Commands for Task 1' in the script.

Example code to highlight grep command

Task 1 case
statement starts
here

Task 1 script
commands were
added here

Task 1 case
statement ends
here

```
mhen90@opus:~/bin
Enter Your Choice: "
read RESPONSE
case $RESPONSE in
1)  # Looking for Love, using grep with r and w options
    echo "Hello $LOGNAME, we are going to look love"
    echo -n "Are you ready? (Hit Enter to continue)"
    read answer
    echo "OK trying your home directory first ..."
    grep -rw love /home/cis90/${LOGNAME%90}
    ;;
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
```

Initial Task 1 script

```
1)      # Looking for Love, using grep with r and w options
        echo "Hello $LOGNAME, we are going to look love"
        echo -n "Are you ready? (Hit Enter to continue)"
        read answer
        echo "OK trying your home directory first ..."
        grep -rw love /home/cis90/${LOGNAME%90}
        ;;
```

Example code to highlight grep command

```
simben90@opus:~/bin
Enter Your Choice: "
read RESPONSE
case $RESPONSE in
  1)    # Looking for Love, using grep with r and w options
        echo "Hello $LOGNAME, we are going to look love"
        echo -n "Are you ready? (Hit Enter to continue)"
        read answer
        echo "OK trying your home directory first ..."
        grep -rw love /home/cis90/${LOGNAME}90
        ;;
  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
```

Implementing all five tasks (6 points each):

- Requirements for each task:

- NO** - Minimum of 10 "original" script command lines
- ✓ - Has comments to explain what it does
- ✓ - 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 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.

Enhanced Task 1 script

```
1)      # Looking for Love, using grep with r and w options
        echo "Hello $LOGNAME, we are going to look love"
        echo -n "Are you ready? (Hit Enter to continue)"
        read answer
        echo "OK trying your home directory first ..."
        grep -rw love /home/cis90/${LOGNAME%90}
        echo "Hello $LOGNAME, lets just list the files now"
        echo -n "Are you ready? (Hit Enter to continue)"
        read answer
        echo "OK here we go ..."
        grep -rwl love /home/cis90/${LOGNAME%90}
        ;;
```


Example code to highlight grep command

```
simben90@opus:~/bin

1) Looking for love in all the wrong places
2) Task 2
3) Task 3
4) Task 4
5) Task 5
6) Exit

Enter Your Choice: "
read RESPONSE
case $RESPONSE in
  1)    # Looking for Love, using grep with
        echo "Hello $LOGNAME, we are going
        echo -n "Are you ready? (Hit Enter
        read answer
        echo "OK trying your home director
        grep -rw love /home/cis90/${LOGNAM
        echo "Hello $LOGNAME, lets just li
        echo -n "Are you ready? (Hit Enter
        read answer
        echo "OK here we go ..."
        grep -rw love /home/cis90/${LOGNA
        ;;
  2)    # Commands for Task 2
-- INSERT --
```

Implementing all five tasks (6 points each):

- Requirements for each task:
- ✓ - Minimum of 10 "original" script command lines
- ✓ - Has comments to explain what it does
- ✓ - 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 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.

Further enhanced Task 1 script

```
1)      # Looking for Love, using grep with r and w options
        echo "Hello $LOGNAME, we are going to look love"
        echo -n "Are you ready? (Hit Enter to continue)"
        read answer
        echo "OK trying your home directory now ..."
        grep -rw love /home/cis90/${LOGNAME%90}
        echo "Hello $LOGNAME, lets just list the files now"
        echo -n "Are you ready? (Hit Enter to continue)"
        read answer
        echo "OK here we go ..."
        grep -rwl love /home/cis90/${LOGNAME%90}
        echo "OK $LOGNAME, lets count those files now"
        echo -n "Are you ready? (Hit Enter to continue)"
        read answer
        echo -n "The number of files containing love is:"
        grep -rwl love /home/cis90/${LOGNAME%90} | wc -l
        ;;
```

Example code to highlight grep command

```
simben90@opus:~/bin
Enter Your Choice: "
read RESPONSE
case $RESPONSE in
  1)    # Looking for Love, using grep with r and w options
        echo "Hello $LOGNAME, we are going to look love"
        echo -n "Are you ready? (Hit Enter to continue)"
        read answer
        echo "OK trying your home directory"
        grep -rw love /home/cis90/${LOGNAME}
        echo "Hello $LOGNAME, lets just list files"
        echo -n "Are you ready? (Hit Enter to continue)"
        read answer
        echo "OK here we go ..."
        grep -rwl love /home/cis90/${LOGNAME}
        echo "OK $LOGNAME, lets count those files"
        echo -n "Are you ready? (Hit Enter to continue)"
        read answer
        echo -n "The number of files containing love is:"
        grep -rwl love /home/cis90/${LOGNAME}
        ;;
  2)    # Commands for Task 2
        ;;
  3)    # Commands for Task 3
        ;;
esac
```

Implementing all five tasks (6 points each):

- Requirements for each task:
- ✓ - Minimum of 10 "original" script command lines
- ✓ - Has comments to explain what it does
- ✓ - 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 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.

Testing your script

```

simmsben@opus:~/bin
CIS 90 Final Project
1) Looking for love in all the wrong places
2) Task 2
3) Task 3
4) Task 4
5) Task 5
6) Exit
Enter Your

simmsben@opus:~/bin
Hello simmsben, we are going to look for love
Are you ready? (Hit Enter to Continue)

simmsben@opus:~/bin
/home/cis90/simmsben/poems/Shakespeare/sonnet26:Lord of my love, to whom in vassalag
e
/home/cis90/simmsben/poems/Shakespeare/sonnet26:      Then may I dare to boast how
I do love t
/home/cis90/
e to any,
/home/cis90/
tle love?
/home/cis90/
ove of me,
/home/cis90/
love of you
/home/cis90/
e"
/home/cis90/
/home/cis90/
/home/cis90/
/home/cis90/
of God. And
Binary file
grep: /home/

I am now goi
Are you read

simmsben@opus:~/bin
/home/cis90/enriqste/poems/Shakespeare/sonnet10:Shall hate be fairer lodg'd then gen
tle love?
/home/cis90/
ove of me,
/home/cis90/
love of yo
/home/cis90/
e"
/home/cis90/
/home/cis90/
/home/cis90/
/home/cis90/
of God. And
Binary file
grep: /home/

I am now goi
Are you read

simmsben@opus:~/bin
Binary file /home/cis90/wattsluk/dogs.tar matches
/home/cis90/wattsluk/cmds: 615 grep "love" sonnet*

simmsben@opus:~/bin
sits
/home/cis90/enriqste/poems/Shakespeare/sonnet3:Of his self-love, to stop posterity?
/home/cis90/enriqste/poems/Shakespeare/sonnet26:Lord of my love, to whom in vassalag
e
/home/cis90/enriqste/poems/Shakespeare/sonnet26:      Then may I dare to boast how
I do love thee,
/home/cis90/enriqste/poems/Shakespeare/sonnet10:For shame deny that thou bear'st lov
e to any,
/home/cis90/enriqste/poems/Shakespeare/sonnet10:Shall hate be fairer lodg'd then gen
tle love?
/home/cis90/enriqste/poems/Shakespeare/sonnet10:      Make thee another self for l
ove of me,
/home/cis90/enriqste/poems/Shakespeare/sonnet15:      And all in war with Time for
love of yo
love of you,
/home/cis90/enriqste/lab07: 741 grep "love" sonnet
/home/cis90/enriqste/lab07: 742 grep "love" sonnet*
/home/cis90/enriqste/cmds: 741 grep "love" sonnet
/home/cis90/enriqste/cmds: 742 grep "love" sonnet*
/home/turnin/cis90/srecklau/lab02:banner "I love you"

I am now goi
Are you read

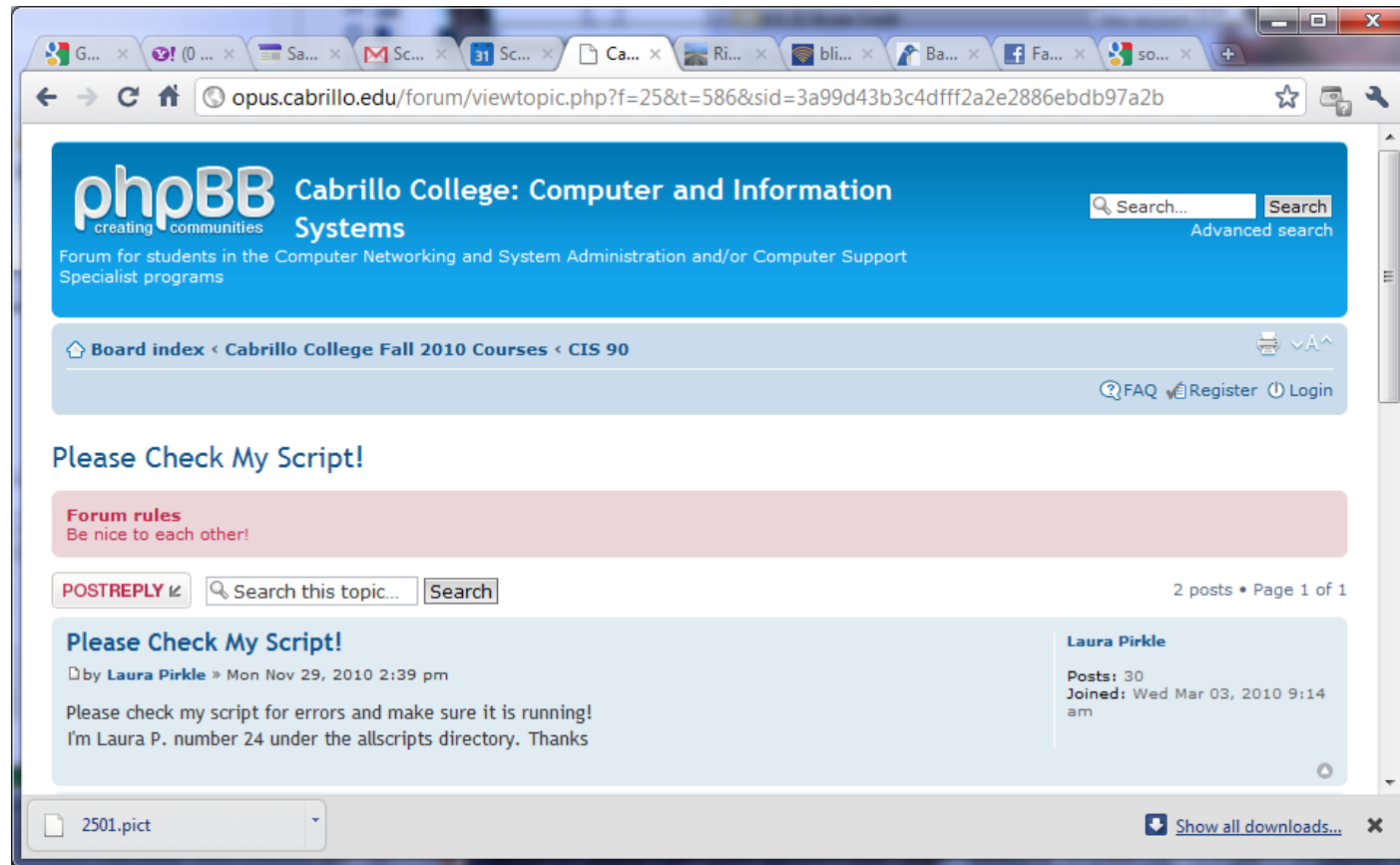
simmsben@opus:~/bin
I am now going to count how many I found
Are you ready? (Hit Enter to Continue)
560

Thats all folks!
Hit the Enter key to return to menu

```

Test your script and make changes till its just the way you want it

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

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

Scripting Tips

vi


```
:%s /oldstring/newstring/g
```

```
rsmms@opus:/home/cis192/depot<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<title>Arwen's CIS 192 Lab 10</title>
</head>
<body>
<h1>Arwen's CIS 192 Lab 10</h1>
<h2>Internet Services</h2>
<div>

</div>
<p>Spring 2009</p>
<div>
<a href="http://validator.w3.org/check/referer"
style="background-color: transparent"

</body>
</html>
~
:%s /Arwen/Elrond/g
```

```
rsmms@opus:/home/cis192/depot<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<title>Elrond's CIS 192 Lab 10</title>
</head>
<body>
<h1>Elrond's CIS 192 Lab 10</h1>
<h2>Internet Services</h2>
<div>

</div>
<p>Spring 2009</p>
<div>
<a href="http://validator.w3.org/check/referer"
style="background-color: transparent"
</a>
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~
</div>
<a href="http://jigsaw.w3.org/css-validator/check/referer"
style="background-color: transparent"
</a>
</div>
</body>
</html>
~
:%s /Arwen/Elrond/g
```




Scripting Tips

`$(cmd)` or ``cmd``

Shell Scripts

Sometimes you want to use the output of a command as an argument to another command

For example:

```
/home/cis90/simben $ find /bin | wc -l  
113
```

```
/home/cis90/simben $ count=`find /bin | wc -l`
```

```
/home/cis90/simben $ echo "There are $count files in /bin"  
There are 113 files in /bin
```

*Using back tics around the command
to evaluate*

Shell Scripts

Sometimes you want to use the output of a command as an argument to another command

For example:

```
/home/cis90/simben $ find /bin | wc -l  
113
```

```
/home/cis90/simben $ count=$(find /bin | wc -l)
```

```
/home/cis90/simben $ echo "There are $count files in /bin"  
There are 113 files in /bin
```

Using `$()` instead of back ticks is an alternate way to do the same thing

Class Exercise

Scripting

Try these commands:

```
banner date
```

```
banner $date
```

```
banner `date`
```

```
banner $(date)
```

Scripting Tips

getting a field from a table

/etc/passwd

```
[rsimms@opus ~]$ cat /etc/passwd
```

< snipped >

```
tumajan:x:1020:103:Janet Tuma:/home/cis90/tumajan:/bin/bash
tumawil:x:1021:103:William Tuma:/home/cis90/tumawil:/bin/bash
kittldou:x:1022:103:Douglas Kittle:/home/cis90/kittldou:/bin/bash
hussabil:x:1023:103:Bilal Hussain:/home/cis90/hussabil:/bin/bash
senantal:x:1024:103:Talley Senanayake:/home/cis90/senantal:/bin/bash
ferrajoe:x:1025:103:Joe Ferrante:/home/cis90/ferrajoe:/bin/bash
balesgeo:x:1026:103:George Bales:/home/cis90/balesgeo:/bin/bash
matliple:x:1029:103:Glen Matlin:/home/cis90/matliple:/bin/bash
krametod:x:1030:103:Todd Kramer:/home/cis90/krametod:/bin/bash
ortegedg:x:1031:103:Edgar Ortega:/home/cis90/ortegedg:/bin/bash
pantogab:x:1032:103:Gabriel Pantoja:/home/cis90/pantogab:/bin/bash
cervajai:x:1033:103:Jaime Cervantes:/home/cis90/cervajai:/bin/bash
donohlin:x:1034:103:Linda Donohue:/home/cis90/donohlin:/bin/bash
mambulie:x:1035:103:Lieven Mambulu:/home/cis90/mambulie:/bin/bash
caseypat:x:1036:103:Patrick Casey:/home/cis90/caseypat:/bin/bash
pragejoe:x:1037:103:Joe Prager:/home/cis90/pragejoe:/bin/bash
rossjun:x:1038:103:Junious Ross:/home/cis90/rossjun:/bin/bash
conydan:x:1039:103:Dan Cony:/home/cis90/conydan:/bin/bash
georgmic:x:1040:103:Michael George:/home/cis90/georgmic:/bin/bash
langlcra:x:1041:103:Craig Langlo:/home/cis90/langlcra:/bin/bash
leekan:x:1042:103:Kang Lee:/home/cis90/leekan:/bin/bash
```

*The ":" serves as the field **delimiter***

The 5th field of each row has the user's first and last name

myscript

```
8)      # Commands for Task 8  
        date  
        ; ;
```

Lets start with something simple

Duke's CIS 90 Final Project

- 1) Color
- 2) My Find Command
- 3) More practice
- 4) Examples - test file attributes
- 5) Examples - simple if statement
- 6) Examples - another if statement
- 7) Examples - logic
- 8) Examples - cut command to get name from /etc/passwd
- 10) Exit

Enter Your Choice: 8

Wed Dec 3 14:00:53 PST 2008

Hit the Enter key to return to menu

myscript

```
8)      # Commands for Task 8
        echo "Hello $LOGNAME"
        date
        ;;
```

*Lets add a Hello with the
users's logname*

Duke's CIS 90 Final Project

- 1) Color
- 2) My Find Command
- 3) More practice
- 4) Examples - test file attributes
- 5) Examples - simple if statement
- 6) Examples - another if statement
- 7) Examples - logic
- 8) Examples - cut command to get name from /etc/passwd
- 10) Exit

Enter Your Choice: 8

Hello roddyduk

Wed Dec 3 14:07:07 PST 2008

Hit the Enter key to return to menu

myscript

```
8)      # Commands for Task 8
        echo "Hello $LOGNAME"
        echo $(cat /etc/passwd | grep $LOGNAME)
        date
        ;;
```

*Now show the
info for the user
in /etc/passwd*

Duke's CIS 90 Final Project

- 1) Color
- 2) My Find Command
- 3) More practice
- 4) Examples - test file attributes
- 5) Examples - simple if statement
- 6) Examples - another if statement
- 7) Examples - logic
- 8) Examples - cut command to get name from /etc/passwd
- 10) Exit

Enter Your Choice: 8

Hello roddyduk

roddyduk:x:1156:103:Duke Roddy:/home/cis90/roddyduk:/bin/bash

Wed Dec 3 14:07:07 PST 2008

Hit the Enter key to return to menu

myscript

```
8)      # Commands for Task 8
        echo "Hello $LOGNAME"
        echo $(cat /etc/passwd | grep $LOGNAME | cut -f5 -d":")
        date
        ; ;
```

*Now cut out the 5th field from the /etc/passwd row using the **cut** command. The **-d** option specifies the delimiter to use.*

Duke's CIS 90 Final Project

- 1) Color
- 2) My Find Command
- 3) More practice
- 4) Examples - test file attributes
- 5) Examples - simple if statement
- 6) Examples - another if statement
- 7) Examples - logic
- 8) Examples - cut command to get name from /etc/passwd
- 10) Exit

Enter Your Choice: 8

Hello roddyduk

Duke Roddy

Wed Dec 3 14:07:07 PST 2008

Hit the Enter key to return to menu

myscript

```
8)      # Commands for Task 8
        echo "Hello $LOGNAME"
        NAME=$(cat /etc/passwd | grep $LOGNAME | cut -f5 -d":" )
        echo "Hello $NAME"
        date
        ;;
```

Same as before, but save the user's name in a variable

Duke's CIS 90 Final Project

- 1) Color
- 2) My Find Command
- 3) More practice
- 4) Examples - test file attributes
- 5) Examples - simple if statement
- 6) Examples - another if statement
- 7) Examples - logic
- 8) Examples - cut command to get name from /etc/passwd
- 10) Exit

Enter Your Choice: 8

Hello roddyduk

Hello Duke Roddy

Wed Dec 3 14:07:07 PST 2008

Hit the Enter key to return to menu

myscript

```
8)      # Commands for Task 8
        echo "Hello $LOGNAME"
        NAME=$(cat /etc/passwd | grep $LOGNAME | cut -f5 -d":" )
        echo "Hello $NAME"
        date
        ;;
```

Get rid of the old Hello \$LOGNAME since we have something better now

```
                Duke's CIS 90 Final Project

1) Color
2) My Find Command
3) More practice
4) Examples - test file attributes
5) Examples - simple if statement
6) Examples - another if statement
7) Examples - logic
8) Examples - cut command to get name from /etc/passwd
10) Exit
```

Enter Your Choice: 8

Hello Duke Roddy

Wed Dec 3 14:07:07 PST 2008

Hit the Enter key to return to menu

myscript

```
8) # Commands for Task 8
echo "Hello $LOGNAME"
NAME=$(cat /etc/passwd | grep $LOGNAME | cut -f5 -d":" | cut -f1 -d" ")
echo "Hello $NAME"
date
;;
```

We can also cut out just the first name using a blank as the delimiter

```
Duke's CIS 90 Final Project
1) Color
2) My Find Command
3) More practice
4) Duke's friend made this one - Thank You
5) Task 5
6) Exit

Enter Your Choice: 8
Hello Duke
Wed Dec  3 14:07:07 PST 2008
Hit the Enter key to return to menu
```

Class Exercise

Make a short script named `example401` that emails a banner of your full name to yourself:

Make a new script in your `bin` directory

`cd bin`

`vi example401`

In `vi` add these lines to your `example401` script then save:

`name=$(cat /etc/passwd | grep $LOGNAME | cut -f5 -d":" )`
`banner $(echo $name) | mail -s "$name" $LOGNAME`

Prepare and run your script

`chmod +x example401`

`example401`

Read your mail to view your new message

`mail`

Scripting Tips

simple if statement

myscript

If statements are used to test if a condition is true and if so execute a specific set of commands

```
5)      # Simple if statement
        echo -n "Enter d or c: "
        read answer
```

Style 1

```
        if [ "$answer" = "d" ]
        then
            date
        fi
```

The date command is executed only if the user typed a "d"

Style 2

```
        if [ "$answer" = "c" ]; then
            cal
        fi

        ;;
```

The cal command is executed only if the user typed a "c"

*Coding styles will vary. Some folks like the **then** on the same line as the **if** followed by a block of indented commands . Some will put **then** on the next line.*

myscript

Duke's CIS 90 Final Project

- 1) Color
- 2) My Find Command
- 3) More practice
- 4) Examples - test file attributes
- 5) Examples - simple if statement
- 6) Examples - logic
- 10) Exit

Enter Your Choice: **5**

Enter d or c: **d**

Sun May 17 10:00:35 PDT 2009

Hit the Enter key to return to menu

```
if [ "$answer" = "d" ]  
then  
    date  
fi
```

The date command runs because \$answer = d

myscript

Duke's CIS 90 Final Project

- 1) Color
- 2) My Find Command
- 3) More practice
- 4) Examples - test file attributes
- 5) Examples - simple if statement
- 6) Examples - logic
- 10) Exit

Enter Your Choice: **5**

Enter d or c: **c**

```
May 2009
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
```

Hit the Enter key to return to menu

```
if [ "$answer" = "c" ]; then
    cal
fi
```

The cal command runs because \$answer = c

Class Exercise

Run the previous example task

- run **allscripts**
- select 50) Duke
- select Task 5 and enter d (for date)
- select Task 5 and enter c (for calendar)

Now look at Duke's code to see how it was done:

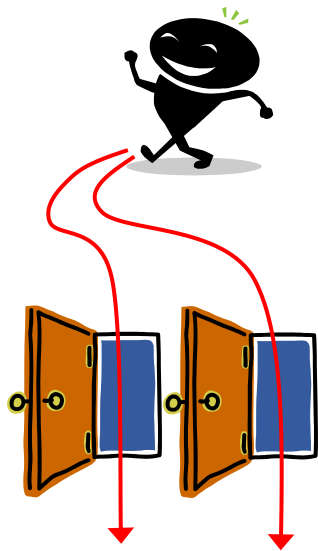
- **vi /home/cis90/rodduk/bin/myscript**

Scripting Tips

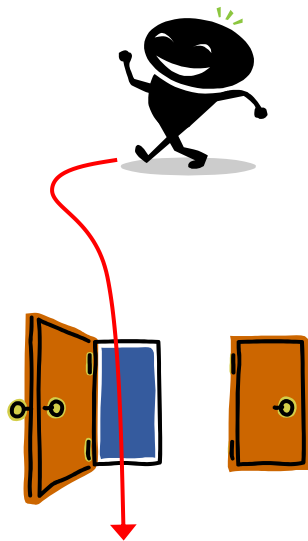
if statement with "or"

p	q	p or q
T	T	T
T	F	T
F	T	T
F	F	F

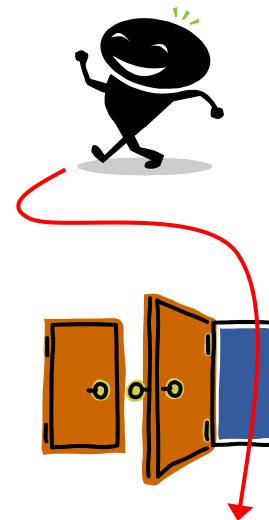
OR logic



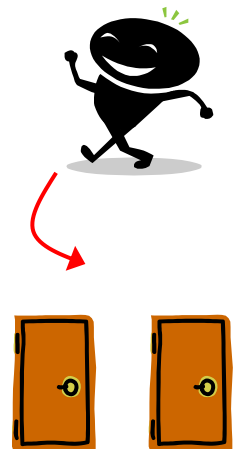
Yes



Yes



Yes



No

myscript

```
6)  # Another if statement  
    echo -n "Enter d or c: "  
    read answer
```

```
    if [ "$answer" = "d" ] || [ "$answer" = "D" ]  
    then  
        date  
    fi
```

*run date if the user
types d or D*

```
    if [ "$answer" = "c" ] || [ "$answer" = "C" ]; then  
        cal  
    fi  
  
;;
```

*run cal if the user
types c or C*

The || is the logical "or" operator

myscript

Duke's CIS 90 Final Project

- 1) Color
- 2) My Find Command
- 3) More practice
- 4) Examples - test file attributes
- 5) Examples - simple if statement
- 6) Examples - another if statement
- 7) Examples - logic
- 10) Exit

Enter Your Choice: **6**

Enter d or c: **d**

Wed May 20 05:07:10 PDT 2009

Hit the Enter key to return to menu

*date is run because user
typed a d*

```
if [ "$answer" = "d" ] || [ "$answer" = "D" ]  
then  
    date  
fi
```

myscript

Duke's CIS 90 Final Project

- 1) Color
- 2) My Find Command
- 3) More practice
- 4) Examples - test file attributes
- 5) Examples - simple if statement
- 6) Examples - another if statement
- 7) Examples - logic
- 10) Exit

Enter Your Choice: **6**

*date is run because user
typed a D*

Enter d or c: **D**

Wed May 20 05:07:38 PDT 2009

Hit the Enter key to return to menu

```
if [ "$answer" = "d" ] || [ "$answer" = "D" ]  
then  
    date  
fi
```


Class Exercise

Make a new script in your bin directory

cd bin

vi example654

In vi add these lines to your script then save:

echo -n "What is your name: "

read answer

if ["\$answer" = "Sylar"] || ["\$answer" = "sylar"]; then

echo "I'm out of here"

fi

Prepare and run your script

chmod +x example654

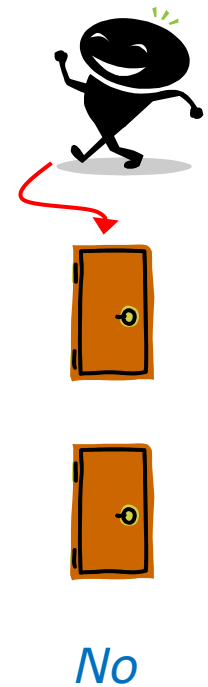
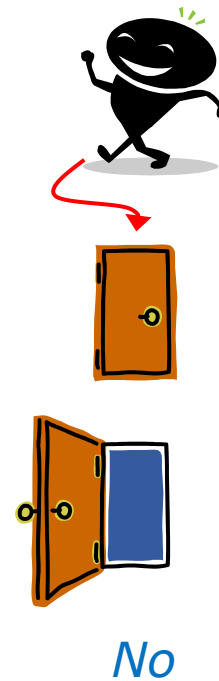
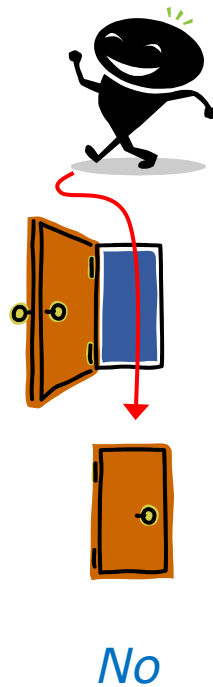
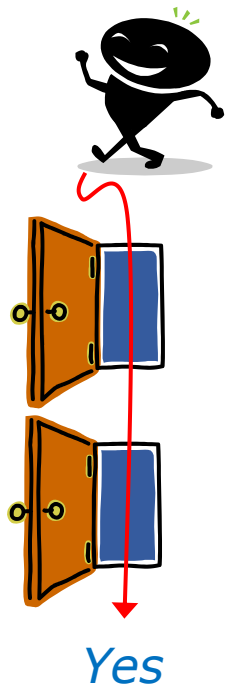
example654

Scripting Tips

if statements with "and"

p	q	p and q
T	T	T
T	F	F
F	T	F
F	F	F

AND logic



myscript

```
6)  # logic example
    echo -n "Is the furnace "on" or off? "
    read furnace
    echo -n "Is there a fire in the fireplace (yes or no)? "
    read fireplace

    if [ "$furnace" = "on" ] && [ "$fireplace" = "yes" ]; then
        echo "It is really hot in here"
    fi

    if [ "$furnace" = "off" ] && [ "$fireplace" = "yes" ]; then
        echo "It is warm and smokey in here"
    fi

    if [ "$furnace" = "on" ] && [ "$fireplace" = "no" ]; then
        echo "It is warm in here"
    fi

    if [ "$furnace" = "off" ] && [ "$fireplace" = "no" ]; then
        echo "It is really freezing in here"
    fi
;;
```

&& means "and"

myscript

Duke's CIS 90 Final Project

- 1) Color
- 2) My Find Command
- 3) More practice
- 4) Examples - test file attributes
- 5) Examples - simple if statement
- 6) Examples - another if statement
- 7) Examples - logic
- 8) Examples - cut command to get name from /etc/passwd
- 10) Exit

Enter Your Choice: **7**

Is the furnace on or off? **off**

Is there a fire in the fireplace (yes or no)? **no**

It is really freezing in here

Hit the Enter key to return to menu

```
if [ "$furnace" = "off" ] && [ "$fireplace" = "no" ]; then  
    echo "It is really freezing in here"  
fi
```

myscript

Duke's CIS 90 Final Project

- 1) Color
- 2) My Find Command
- 3) More practice
- 4) Examples - test file attributes
- 5) Examples - simple if statement
- 6) Examples - another if statement
- 7) Examples - logic
- 8) Examples - cut command to get name from /etc/passwd
- 10) Exit

Enter Your Choice: **7**

Is the furnace on or off? **on**

Is there a fire in the fireplace (yes or no)? **no**

It is warm in here

Hit the Enter key to return to menu

```
if [ "$furnace" = "on" ] && [ "$fireplace" = "no" ]; then
    echo "It is warm in here"
fi
```



Class Exercise

Run the previous example task

- run **allscripts**
- select 50) Duke
- select Task 7 several times with different answers

Now look at Duke's code to see how it was done:

- **vi /home/cis90/rodduk/bin/myscript**

Scripting Tips

if file types

myscript

```
4) # More example IF statments
    echo "The files in this directory are: "
    ls -l
    echo -n "Which file are you interested in? : "
    read filename
```

```
    echo "Here are some details about $filename:"
    file $filename
```

*tests to see
if it's a
regular file*

```
    if [ -f $filename ]; then
        echo $filename is a regular file
        echo "Here is long listing of the $filename" file:
        ls -l $filename
    fi
```

*tests to see
if it's a
directory*

```
    if [ -d $filename ]; then
        echo $filename is a directory
        echo "Here is a long listing of the $filename directory:"
        ls -ld $filename
    fi
;;
```

myscript

Duke's CIS 90 Final Project

- 1) Color
- 2) My Find Command
- 3) More practice
- 4) Task 4
- 5) Task 5
- 6) Exit

Enter Your Choice: **4**

The files in this directory are:

1976.egg

Anon

Blake

Shakespeare

Yeats

Which file are you interested in? : **1976.egg**

Here are some details about 1976.egg:

1976.egg: ASCII English text, with escape sequences

1976.egg is a regular file

Here is long listing of the 1976.egg file:

-rw-r--r-- 1 squid squid 734 Apr 8 10:01 1976.egg

Hit the Enter key to return to menu

myscript

Duke's CIS 90 Final Project

- 1) Color
- 2) My Find Command
- 3) More practice
- 4) Task 4
- 5) Task 5
- 6) Exit

Enter Your Choice: **4**

The files in this directory are:

1976.egg

Anon

Blake

Shakespeare

Yeats

Which file are you interested in? : **Anon**

Here are some details about Anon:

Anon: directory

Anon is a directory

Here is a long listing of the Anon directory:

drwxr-xr-x 2 roddyduk cis90 4096 Apr 8 10:01 Anon

Hit the Enter key to return to menu

-d file = True if the file exists and is a directory.
-e file = True if the file exists.
-f file = True if the file exists and is a regular file
-k file = True if the files' "sticky" bit is set.
-L file = True if the file exists and is a symbolic link.
-r file = True if the file exists and is readable.
-s file = True if the file exists and is not empty.
-u file = True if the file exists and its set-user-id bit is set.
-w file = True if the file exists and is writable.
-x file = True if the file exists and is executable.
-O file = True if the file exists and is owned by the effective user id.
-G file = True if the file exists and is owned by the effective group id.
file1 -nt file2 = True if file1 is newer, by modification date, than file2.
file1 -ot file2 = True if file1 is older than file2.

Class Exercise

Run the previous example task

- run **allscripts**
- select 50) Duke
- select Task 4

Now look at Duke's code to see how it was done:

- **vi /home/cis90/rodduk/bin/myscript**

Scripting Tips

if then else statement

myscript

```
3)      # Commands for Task 3
        NAME=$(cat /etc/passwd | grep $LOGNAME | cut -f5 -d":" )
        echo "Hello $NAME"
        date '+%A'
        date '+%A, %B %d, %Y'
        ;;
```

Duke's CIS 90 Final Project

- 1) Color
- 2) My Find Command
- 3) More practice
- 4) Duke's friend made this one - Thank You
- 5) Task 5
- 6) Exit

Enter Your Choice: 3

Hello Duke Roddy

Wednesday

Wednesday, December 03, 2008

Hit the Enter key to return to menu

myscript

```
3)      # Commands for Task 3
        NAME=$(cat /etc/passwd | grep $LOGNAME | cut -f5 -d":" )
        echo "Hello $NAME"
        echo "$NAME, Do you like short or long dates?"
        echo -n "Enter 1 for short or 2 for long: "
        read ANSWER
        if [ "$ANSWER" = 1 ]; then
            date '+%A'
        else
            date '+%A, %B %d, %Y'
        fi
        ;;
```

```
Enter Your Choice: 3
Hello Duke Roddy
Duke Roddy, Do you like short or long dates?
Enter 1 for short or 2 for long: 1
Wednesday
Hit the Enter key to return to menu
```

```
Enter Your Choice: 3
Hello Duke Roddy
Duke Roddy, Do you like short or long dates?
Enter 1 for short or 2 for long: 2
Wednesday, December 03, 2008
Hit the Enter key to return to menu
```


Scripting Tips

Using the set command

```
[rsimms@opus scripts]$ set dogs cats birds humans
```

```
[rsimms@opus scripts]$ echo $1  
dogs
```

```
[rsimms@opus scripts]$ echo $2  
cats
```

```
[rsimms@opus scripts]$ echo $3  
birds
```

```
[rsimms@opus scripts]$ echo $4  
humans
```

```
[rsimms@opus scripts]$ echo $#  
4
```

```
[rsimms@opus scripts]$ echo $*  
dogs cats birds humans
```

The set command parses the arguments it receives.

\$1 is set to the first argument, \$2 is set to the second argument and so forth.

is set to the total number of arguments.

```
[rsimms@opus bin]$ echo $(ls)
```

```
1975.egg app banner datecal enlightenment hi I myscript  
myscript.rododyduk myscript.v1 newscript old program quiet quiet.bak  
script treed tryme typescript zoom
```

```
[rsimms@opus bin]$ set $(ls)
```

```
[rsimms@opus bin]$ echo $3  
banner
```

The set command parses the arguments it receives.

```
[rsimms@opus bin]$ echo $7  
I
```

*\$1 is set to the first argument,
\$2 is set to the second argument and so forth.*

```
[rsimms@opus bin]$ echo $11  
1975.egg1
```

\$# is set to the total number of arguments.

```
[rsimms@opus bin]$ echo $#  
20
```

```
[rsimms@opus bin]$ echo "The fifth file in this directory is $5"  
The fifth file in this directory is enlightenment  
[rsimms@opus bin]$
```

```
[rsimms@opus scripts]$ finger $LOGNAME
```

```
Login: rsimms                               Name: Rich Simms
Directory: /home/rsimms                     Shell: /bin/bash
On since Mon May 18 14:38 (PDT) on pts/1 from 207.62.186.30
Mail last read Mon May 18 16:09 2009 (PDT)
No Plan.
```

```
[rsimms@opus scripts]$ finger $LOGNAME | head -1
```

```
Login: rsimms                               Name: Rich Simms
```

```
[rsimms@opus scripts]$ set $(finger $LOGNAME | head -1)
```

```
[rsimms@opus scripts]$ echo $1
```

```
Login:
```

```
[rsimms@opus scripts]$ echo $2
```

```
rsimms
```

```
[rsimms@opus scripts]$ echo $3
```

```
Name:
```

```
[rsimms@opus scripts]$ echo $4
```

```
Rich
```

```
[rsimms@opus scripts]$ echo $5
```

```
Simms
```

```
[rsimms@opus scripts]$ firstname=$4
```

```
[rsimms@opus bin]$ echo My first name is $firstname
```

```
My first name is Rich
```

The set command parses the arguments it receives. \$1 is set to the first argument, \$2 is set to the second argument and so forth. \$# is set to the total number of arguments.

Class Exercise

Make a new script in your bin directory

cd bin

vi example777

In vi add these lines to your script then save:

set \$(finger \$LOGNAME | head -1)

firstname=\$4

echo My first name is \$firstname

Prepare and run your script

chmod +x example777

example777



Scripting Tips

color

Using Color

Black 0;30

Dark Gray 1;30

Blue 0;34

Light Blue 1;34

Green 0;32

Light Green 1;32

Cyan 0;36

Light Cyan 1;36

Red 0;31

Light Red 1;31

Purple 0;35

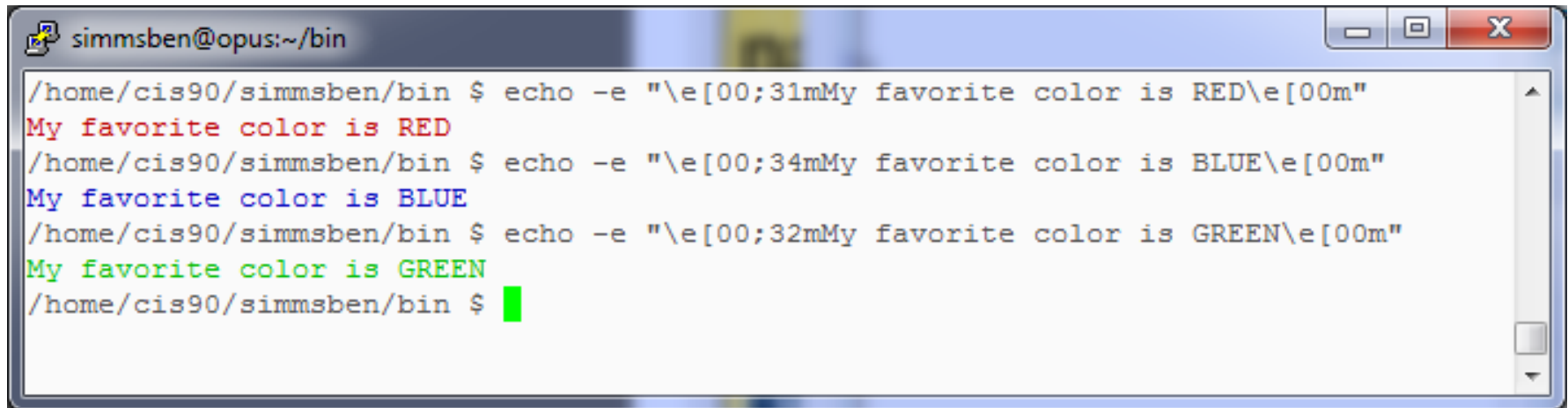
Light Purple 1;35

Brown 0;33

Yellow 1;33

Light Gray 0;37

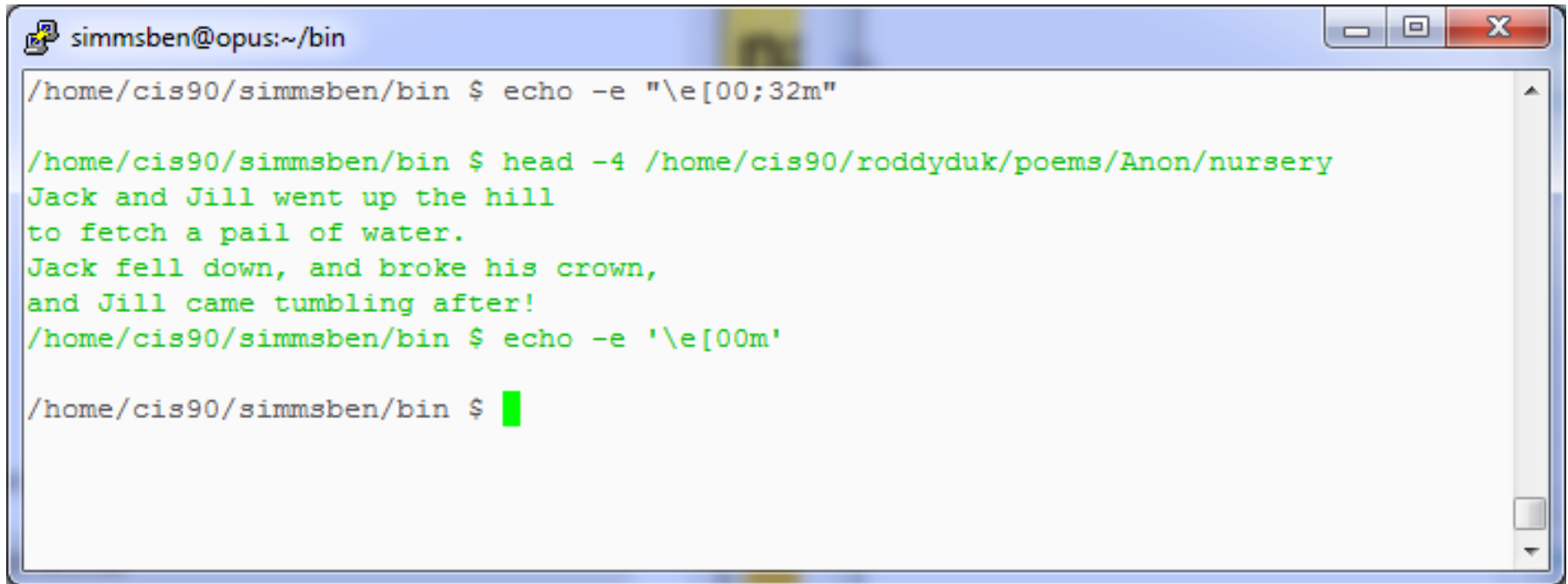
White 1;37

A terminal window titled 'simmsben@opus:~/bin' showing three commands and their color-coded outputs. The first command uses '\e[00;31m' to print 'My favorite color is RED' in red. The second uses '\e[00;34m' to print 'My favorite color is BLUE' in blue. The third uses '\e[00;32m' to print 'My favorite color is GREEN' in green. The prompt is a green block character.

```
simmsben@opus:~/bin
/home/cis90/simmsben/bin $ echo -e "\e[00;31mMy favorite color is RED\e[00m"
My favorite color is RED
/home/cis90/simmsben/bin $ echo -e "\e[00;34mMy favorite color is BLUE\e[00m"
My favorite color is BLUE
/home/cis90/simmsben/bin $ echo -e "\e[00;32mMy favorite color is GREEN\e[00m"
My favorite color is GREEN
/home/cis90/simmsben/bin $ █
```

Make sure to use echo -e to enable interpretation of backslash escapes

Using Color

A terminal window titled 'simmsben@opus:~/bin' with standard window controls. The terminal shows a sequence of commands and their outputs. The first command 'echo -e "\e[00;32m"' results in a blank line. The second command 'head -4 /home/cis90/roddyduk/poems/Anon/nursery' outputs four lines of text in green. The third command 'echo -e '\e[00m'' results in a blank line. The prompt for the next command is shown with a green cursor.

```
simmsben@opus:~/bin
/home/cis90/simmsben/bin $ echo -e "\e[00;32m"

/home/cis90/simmsben/bin $ head -4 /home/cis90/roddyduk/poems/Anon/nursery
Jack and Jill went up the hill
to fetch a pail of water.
Jack fell down, and broke his crown,
and Jill came tumbling after!
/home/cis90/simmsben/bin $ echo -e '\e[00m'

/home/cis90/simmsben/bin $ █
```

Use `echo -e '\e[00m'` to revert to normal



Scripting Tips

home directories and user names

Going from CIS 90 home directory name to username

```
/home/cis90/simben $ echo $HOME  
/home/cis90/simben
```

```
/home/cis90/simben $ basename $HOME  
simben
```

*The **basename** command
extracts the filename from the
end of a pathname*

```
/home/cis90/simben $ echo $(basename $HOME)  
simben
```

```
/home/cis90/simben $ echo $(basename $HOME) 90  
simben90
```

*This is how you tack 90
on to the home directory
filename*

```
/home/cis90/simben $ userid=`echo $(basename $HOME) 90`  
/home/cis90/simben $ echo The home directory of $userid is $HOME  
The home directory of simben90 is /home/cis90/simben
```

Going from CIS 90 home directory name to username

```
/home/cis90/simben $ finger $(basename $HOME) 90
Login: simben90                      Name: Benji Simms
Directory: /home/cis90/simben        Shell: /bin/bash
On since Wed May 16 08:09 (PDT) on pts/2 from 50-0-68-
235.dsl.dynamic.fusionbroadband.com
No mail.
Plan:
To pass this course with flying colors!
```

*Determining the username from the home directory name and then using it as an argument to the **finger** command*

Going from CIS 90 username to home directory name

```
/home/cis90/simben $ echo $LOGNAME  
simben90
```

This variable holds your username

```
/home/cis90/simben $ echo ${LOGNAME%90}  
simben
```

This is how you strip text off the end of a string

```
/home/cis90/simben $ file=`echo ${LOGNAME%90}`  
/home/cis90/simben $ echo $file  
simben
```

This sets a new variable named file to hold the filename

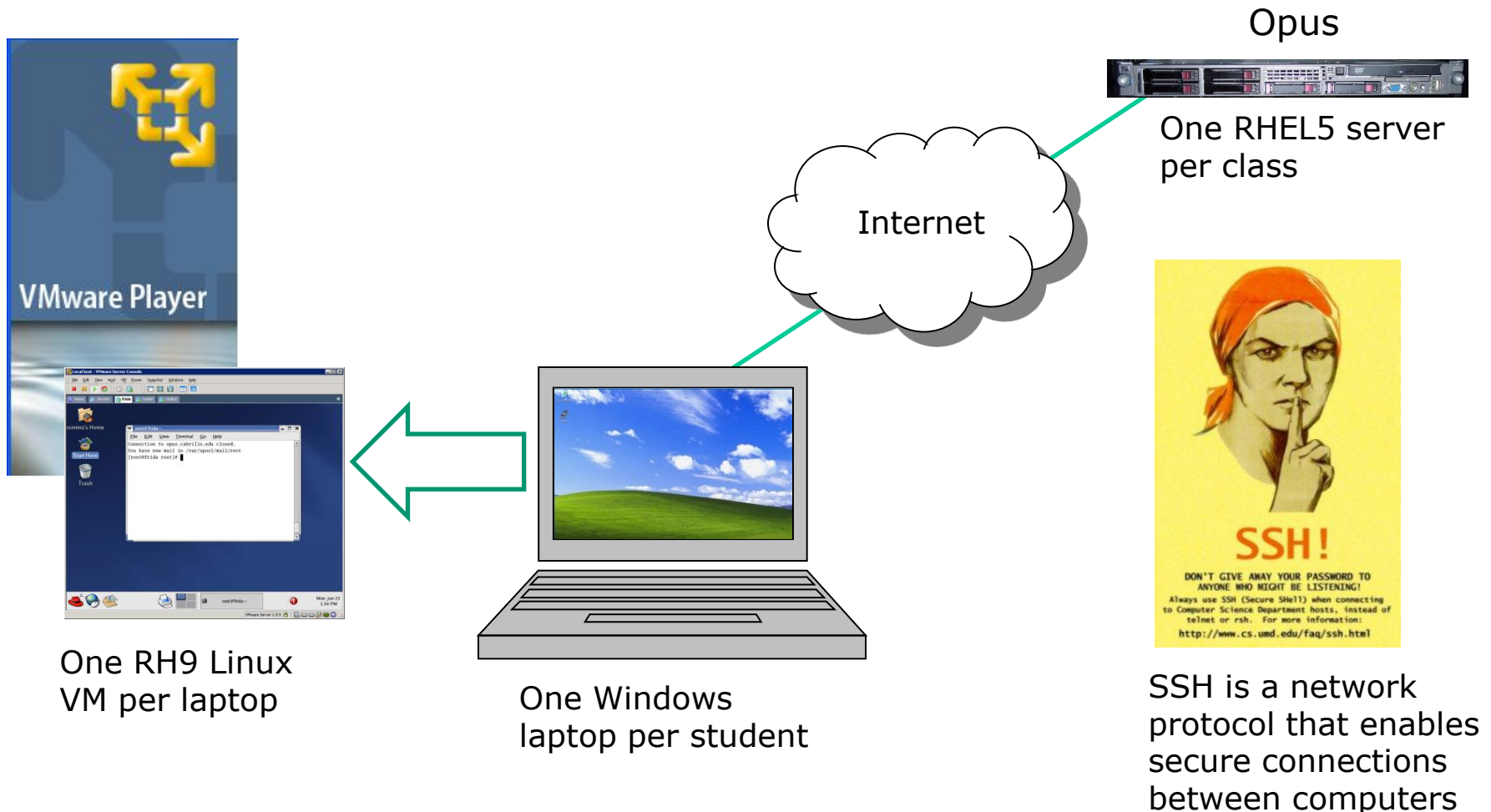
```
/home/cis90/simben $ echo The home of $LOGNAME is /home/cis90/$file  
The home of simben90 is /home/cis90/simben
```

And this is how you could use it

scp

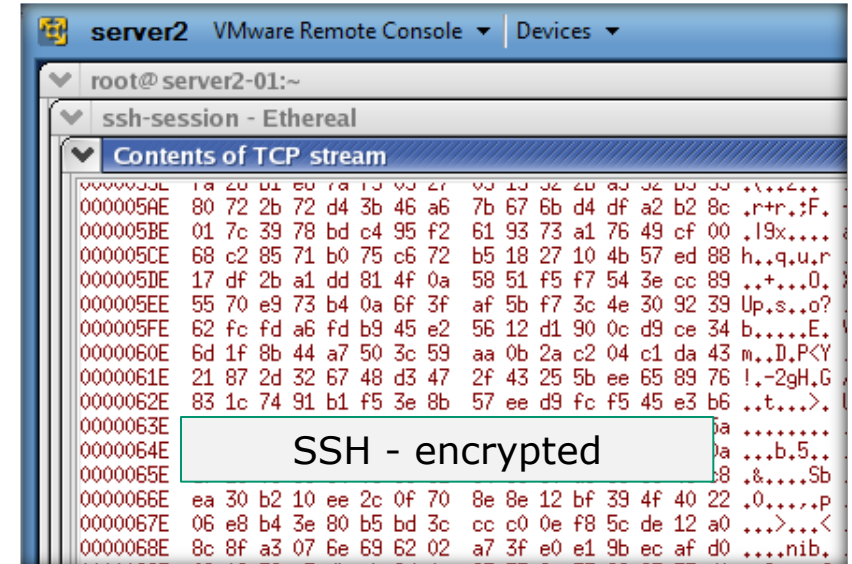
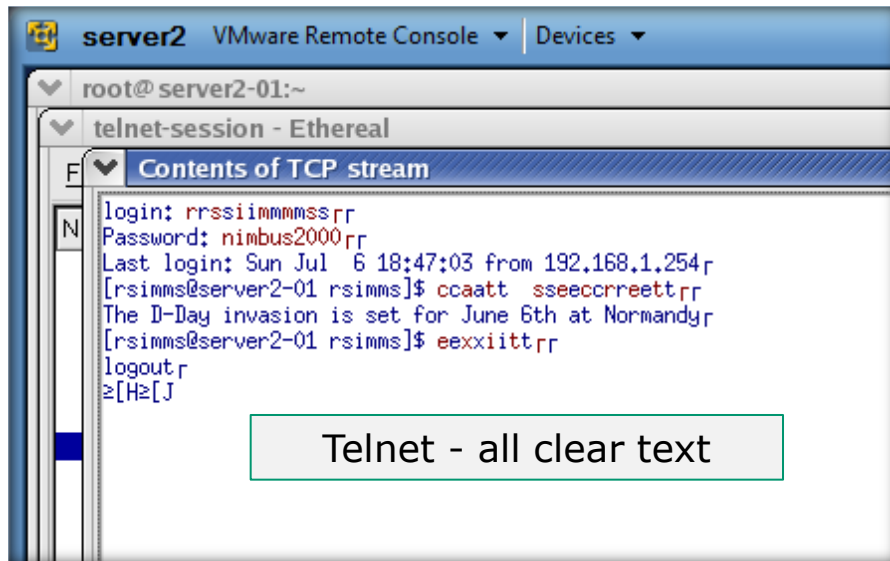
Copying your files on Opus to
another Linux system

Classroom PC's, VMs and Remote Server



Telnet and SSH (Secure Shell)

Opus



Sniffer view of a Telnet session

Sniffer view of a SSH session



Local computer

ssh protocol

Secure Shell Protocol

- Allows secure (encrypted connections between computers)
 - ssh command – secure login and terminal sessions
 - scp command – secure file copies between computers

scp

Copy commands **copy file(s)** to a **Destination**

- cp

- copies files on the same computer
- examples:

```
cp /etc/hosts myhosts  
cp riddle1 riddles/  
cp /home/cis90/bin/tally .
```

*For the **cp** command
each argument is a
pathname*

- scp

- copies files between computers:
- examples:

```
scp simben90@opus.cabrillo.edu:/etc/hosts myhosts  
scp riddle23 simben90@opus.cabrillo.edu:riddles  
scp simben90@opus.cabrillo.edu:bin/myscript .
```

*For the **scp** command, arguments for remote files must include
username, hostname and pathname*

scp

Remote

Local

scp

simben90@opus.cabrillo.edu:bin/myscript

.

Copy the file `myscript` from `simben90`'s home `bin/` directory on the remote system `Opus` to "here"

scp example

Copying project file on Opus to local Linux system

*use @ with no spaces
to delimit username
from hostname*

*use : with no spaces to
delimit hostname from
pathname*

Remote

Local

scp

simben90@opus.cabrillo.edu:bin/myscript

.

Relative or absolute
pathname.

Either the IP address or hostname of the remote
computer. Needed for connection over the Internet.

The username on the remote computer. Needed for
authentication and to establish the home directory on remote
system

Copying a file from Opus to Sun

On Opus

```
/home/cis90/simben $ ls -l ../depot/scrooge
-rw-r--r-- 1 rsimms cis90 33939 Feb 27 10:11 ../depot/scrooge
/home/cis90/simben $
/home/cis90/simben $ head -3 ../depot/scrooge
Stave 2: The First of the Three Spirits
```

```
When Scrooge awoke, it was so dark, that looking out of bed, he could
/home/cis90/simben $
```

On Sun

```
[simben90@sun ~] $ scp simben90@opus.cabrillo.edu:../depot/scrooge .
simben90@opus.cabrillo.edu's password:
scrooge                                     100%   33KB   33.1KB/s   00:00
[simben90@sun ~] $ head -3 scrooge
Stave 2: The First of the Three Spirits
```

```
When Scrooge awoke, it was so dark, that looking out of bed, he could
[simben90@sun ~] $
```

Copying multiple files from Opus to Sun

On Opus

```
/home/cis90/simben $ ls ../depot/random*  
../depot/randomints  ../depot/randomjokes  ../depot/randomwords  
/home/cis90/simben $
```

On Sun

```
[simben90@sun ~] $ scp simben90@opus.cabrillo.edu:../depot/random* mydepot/  
simben90@opus.cabrillo.edu's password:  
randomints                100% 2449      2.4KB/s   00:00  
randomjokes               100%  272      0.3KB/s   00:00  
randomwords               100%  16KB    15.7KB/s   00:00  
[simben90@sun ~] $
```

Copying a file from Sun to Opus

On Sun

```
[simben90@sun ~] $ echo "I love Linux" > file25
[simben90@sun ~] $ cat file25
I love Linux
[simben90@sun ~] $ scp file25 simben90@opus.cabrillo.edu:
simben90@opus.cabrillo.edu's password:
file25                                     100%   13      0.0KB/s   00:00
[simben90@sun ~] $
```

On Opus

```
/home/cis90/simben $ ls -l file25
-rw-r--r-- 1 simben90 cis90 13 May 15 17:56 file25
/home/cis90/simben $ cat file25
I love Linux
/home/cis90/simben $
```

Copying a file from Sun to Opus

On Sun

```
[simben90@sun ~] $ echo I love UNIX > file99
[simben90@sun ~] $ scp file99 simben90@opus.cabrillo.edu:iloveunix
simben90@opus.cabrillo.edu's password:
file99                                     100%   12      0.0KB/s   00:00
[simben90@sun ~] $
```

On Opus

```
/home/cis90/simben $ ls iloveunix
iloveunix
/home/cis90/simben $ cat iloveunix
I love UNIX
/home/cis90/simben $
```

Copying multiple files from Sun to Opus

On Sun

```
[simben90@sun ~] $ ls Poems/[MD]*
```

```
Poems/Dickinson:
```

```
bird  dogtail  drop  light
```

```
Poems/Millay:
```

```
blueflag  singing  weaver
```

```
[simben90@sun ~] $
```

```
[simben90@sun ~] $ scp -r Poems/[MD]* simben90@opus.cabrillo.edu:poems/
```

```
simben90@opus.cabrillo.edu's password:
```

dogtail	100%	424	0.4KB/s	00:00
bird	100%	572	0.6KB/s	00:00
light	100%	547	0.5KB/s	00:00
drop	100%	484	0.5KB/s	00:00
singing	100%	1800	1.8KB/s	00:00
weaver	100%	448	0.4KB/s	00:00
blueflag	100%	5432	5.3KB/s	00:00

```
[simben90@sun ~] $
```


tar

tar command

tar *options tarfile files*

To simplify file transfers, Windows users typically “zip” multiple files together into a single “zipfile”.

Linux users use the **tar** command to do this and “archive” multiple files into a single “tarball”.

tar command

```
tar cvf tarfile pathname
```

*c = **create***

v = verbose

f = filename (which must immediately follow)

```
tar tvf tarfile
```

*t = table of contents (to **view** files in a archive)*

v = verbose

f = filename (which must immediately follow)

```
tar xvf tarfile
```

*x = **extract** files in archive*

v = verbose

f = filename (which must immediately follow)

tar command

Create a tarball out of our local bin directory

```
/home/cis90/simben $ tar cvf cis90scripts.tar bin
```

```
bin/
```

```
bin/tumajan
```

```
bin/banner
```

```
bin/enlightenment
```

```
bin/hi
```

```
bin/script14
```

```
bin/myscript
```

```
bin/tryme
```

```
bin/debug/
```

```
bin/debug/files.simben90
```

```
bin/debug/hovdav01
```

```
bin/debug/ahrmat2
```

```
bin/script99
```

```
bin/datecal
```

```
bin/treed
```

```
bin/rossjun
```

```
bin/zoom
```

```
bin/I
```

```
bin/app
```

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

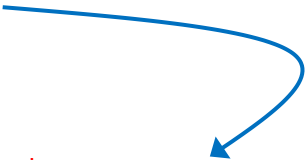
*relative pathname to
directory to archive*

name of archive file

tar command

Tarballs show as red in listings

```
/home/cis90/simben $ ls -ld c*  
-rw-rw-r-- 1 simben90 cis90 71680 May 16 09:38 cis90scripts.tar  
drwx--x--- 4 simben90 cis90 4096 Apr 5 07:52 class  
-rw-r--r-- 1 simben90 cis90 2900 Mar 26 14:28 cmds  
-rw-r--r-- 1 simben90 cis90 0 Apr 5 07:40 critter1  
-rw-r--r-- 1 simben90 cis90 0 Apr 5 07:40 critter2  
/home/cis90/simben $
```



tar command

View contents of a tarball

```
/home/cis90/simben $ tar tvf cis90scripts.tar
drwxr-x--- simben90/cis90      0 2012-05-15 17:23:07 bin/
-rwxr-xr-x simben90/cis90 12854 2012-04-30 08:29:38 bin/tumajan
-r-xr-x--- simben90/cis90  6160 2003-08-28 22:39:53 bin/banner
-r-xr-xr-- simben90/cis90  3442 2012-02-04 16:36:52 bin/enlightenment
-r-xr-x--- simben90/cis90   107 2001-07-20 21:06:07 bin/hi
-rwxr-x--- simben90/cis90  1221 2012-05-15 15:07:03 bin/script14
-rwxr-x--- simben90/cis90 10490 2012-05-08 14:21:06 bin/myscript
-r-xr-x--- simben90/cis90   174 2004-03-04 13:02:58 bin/tryme
drwxrwxr-x simben90/cis90      0 2012-05-15 17:06:03 bin/debug/
-rw-r--r-- simben90/cis90 10567 2012-05-15 16:26:18 bin/debug/files.simben90
-rwxrwxr-x simben90/cis90   236 2012-05-06 19:28:22 bin/debug/hovdav01
-rwxr-x--- simben90/cis90  4828 2012-05-15 16:55:35 bin/debug/ahrmata2
-rwxrwxr-x simben90/cis90   250 2012-05-06 08:59:29 bin/script99
-r-xr-x--- simben90/cis90   509 2002-06-06 14:59:53 bin/datecal
-r-xr-x--- simben90/cis90   190 2001-07-20 15:04:39 bin/treed
-rwxr-xr-x simben90/cis90  3748 2012-04-30 08:29:58 bin/rossjun
-r-xr-x--- simben90/cis90    74 2001-07-20 15:18:06 bin/zoom
-r-xr-x--- simben90/cis90   375 2003-10-20 18:36:41 bin/I
-r-xr-x--- simben90/cis90   220 2004-04-22 18:51:38 bin/app
/home/cis90/simben $
```

tar command

On another Linux system (Sun in Vlab)

```
[simben90@sun ~]$ ls -l bin
total 36
-rwxr-xr-x. 1 simben90 cis90  220 Apr 22  2004 app
-rwxr-xr-x. 1 simben90 cis90 6160 Aug 28  2003 banner
-rwxr-xr-x. 1 simben90 cis90 3391 Feb 20 11:32 enlightenment
-rwxr-xr-x. 1 simben90 cis90  107 Jul 20  2001 hi
-rwxr-x--x. 1 simben90 cis90  375 Oct 20  2003 I
-rwxr-xr-x. 1 simben90 cis90  190 Jul 20  2001 treed
-rwxr-xr-x. 1 simben90 cis90  174 Mar  4  2004 tryme
-rwxr-xr-x. 1 simben90 cis90   74 Jul 20  2001 zoom
[simben90@sun ~]$
```

tar command

On another Linux system (Sun in Vlab)

```
[simben90@sun ~]$ scp simben90@opus.cabrillo.edu:cis90* .  
simben90@opus.cabrillo.edu's password:  
cis90scripts.tar
```

*Copy tarball
from Opus*

100% 70KB 70.0KB/s 00:00

```
[simben90@sun ~]$ tar xvf cis90scripts.tar
```

Extract tarball files on Sun

```
bin/  
bin/tumajan  
bin/banner  
bin/enlightenment  
bin/hi  
bin/script14  
bin/myscript  
bin/tryme  
bin/debug/  
bin/debug/files.simben90  
bin/debug/hovdav01  
bin/debug/ahrmat2  
bin/script99  
bin/datecal  
bin/treed  
bin/rossjun  
bin/zoom  
bin/I  
bin/app  
[simben90@sun ~]
```

*Be careful, this will
overwrite any files with
the same name*

tar command

After extraction

```
[simben90@sun ~]$ ls -l bin
total 84
-r-xr-x---. 1 simben90 cis90   220 Apr 22  2004 app
-r-xr-x---. 1 simben90 cis90  6160 Aug 28  2003 banner
-r-xr-x---. 1 simben90 cis90   509 Jun  6  2002 datecal
drwxr-xr-x. 2 simben90 cis90  4096 May 15 17:06 debug
-r-xr-xr--. 1 simben90 cis90  3442 Feb  4 16:36 enlightenment
-r-xr-x---. 1 simben90 cis90   107 Jul 20  2001 hi
-r-xr-x---. 1 simben90 cis90   375 Oct 20  2003 I
-rwxr-x---. 1 simben90 cis90 10490 May  8 14:21 myscript
-rwxr-xr-x. 1 simben90 cis90   3748 Apr 30 08:29 rossjun
-rwxr-x---. 1 simben90 cis90   1221 May 15 15:07 script14
-rwxr-xr-x. 1 simben90 cis90    250 May  6 08:59 script99
-r-xr-x---. 1 simben90 cis90   190 Jul 20  2001 treed
-r-xr-x---. 1 simben90 cis90   174 Mar  4  2004 tryme
-rwxr-xr-x. 1 simben90 cis90 12854 Apr 30 08:29 tumajan
-r-xr-x---. 1 simben90 cis90    74 Jul 20  2001 zoom
[simben90@sun ~]$
```

Class Exercise

tar

On Opus, tar up you entire home directory. Use your own username to name the tarball:

```
cd    start in your home directory  
tar cvf cis90bin.tar bin  
tar tvf cis90bin.tar
```

On sun, copy your tarfile from Opus and extract it

```
scp $LOGNAME@opus.cabrillo.edu:cis90* .  
tar xvf cis90bin.tar
```

Wrap up

Commands:

basename

scp

tar

if then else

[]

- extract filename from pathname
- secure copy command
- archive command
- conditionals in scripts
- for logic tests in scripts

Next Class

**Project is due
next week!**

Backup

Class Exercise

- Start up the local VM named Frodo and login as cis90
- Copy the files banner and myscript from your bin directory on Opus

```
scp logname@opus.cabrillo.edu:bin/myscript .
```

```
scp logname@opus.cabrillo.edu:bin/banner .
```

- Create a bin directory on your local VM (if needed)

```
mkdir bin
```

- Move banner and myscript to your new bin directory

```
mv banner myscript bin/
```

- Check permissions to make sure myscript and banner have execute permissions

```
cd bin
```

```
ls -l
```

- Run the scripts you copied from Opus:

```
./myscript
```

```
./banner "Hello"
```

Class Exercise

scp

- Copy all the files and directories in your Opus bin directory to your local bin directory

```
scp -r logname@opus.cabrillo.edu:bin/* .
```

- Now see if you can copy your entire poems directory on Opus to a new poems directory in your local cis90 home directory.