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

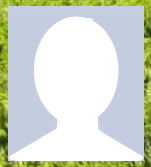
- CCC Confer wall paper -

- Materials uploaded -
- Backup slides, CCC info, handouts on flash drive -
- Check that room headset is charged Aptos (backup) -

CIS 90 - Lesson 11

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


Sean C.

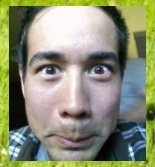

Don

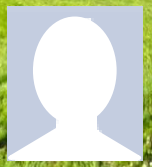

Carlile


Andrew


Sean Fa.


Carter


Sean Fy.


Dajan

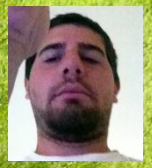

Bryn


Rita


Kelly


Ben


Ray


Michael

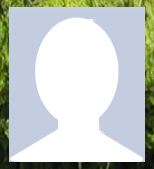

Evan


Josh


Carlos


Gustavo


Jessica


Evie

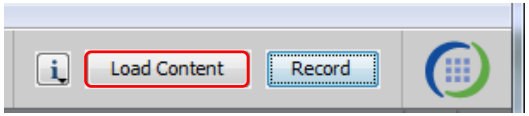

Jacob


Humberto

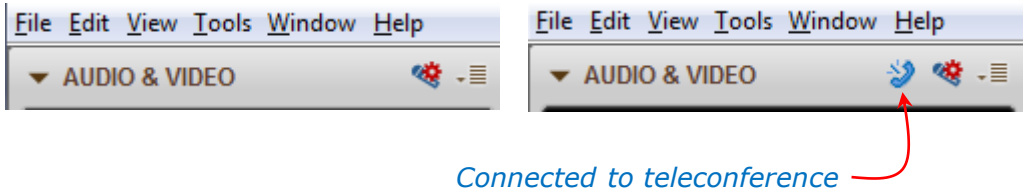

Chad

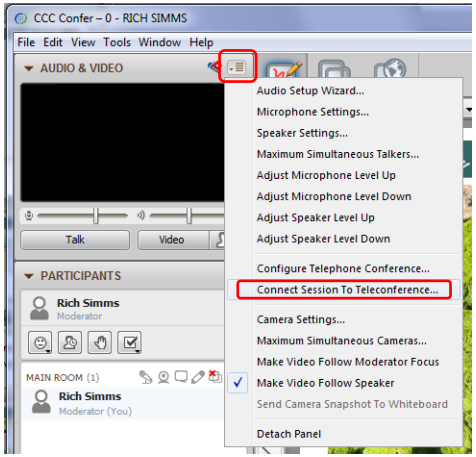
Email me (risimms@cabrillo.edu) a relatively current photo of your face for 3 points extra credit

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

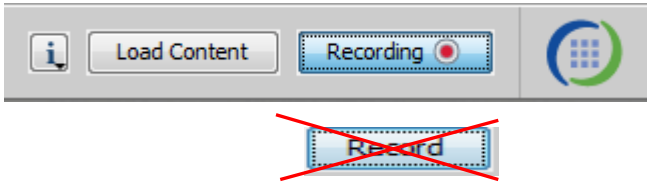


[ ] Connect session to Teleconference





[ ] Is recording on?

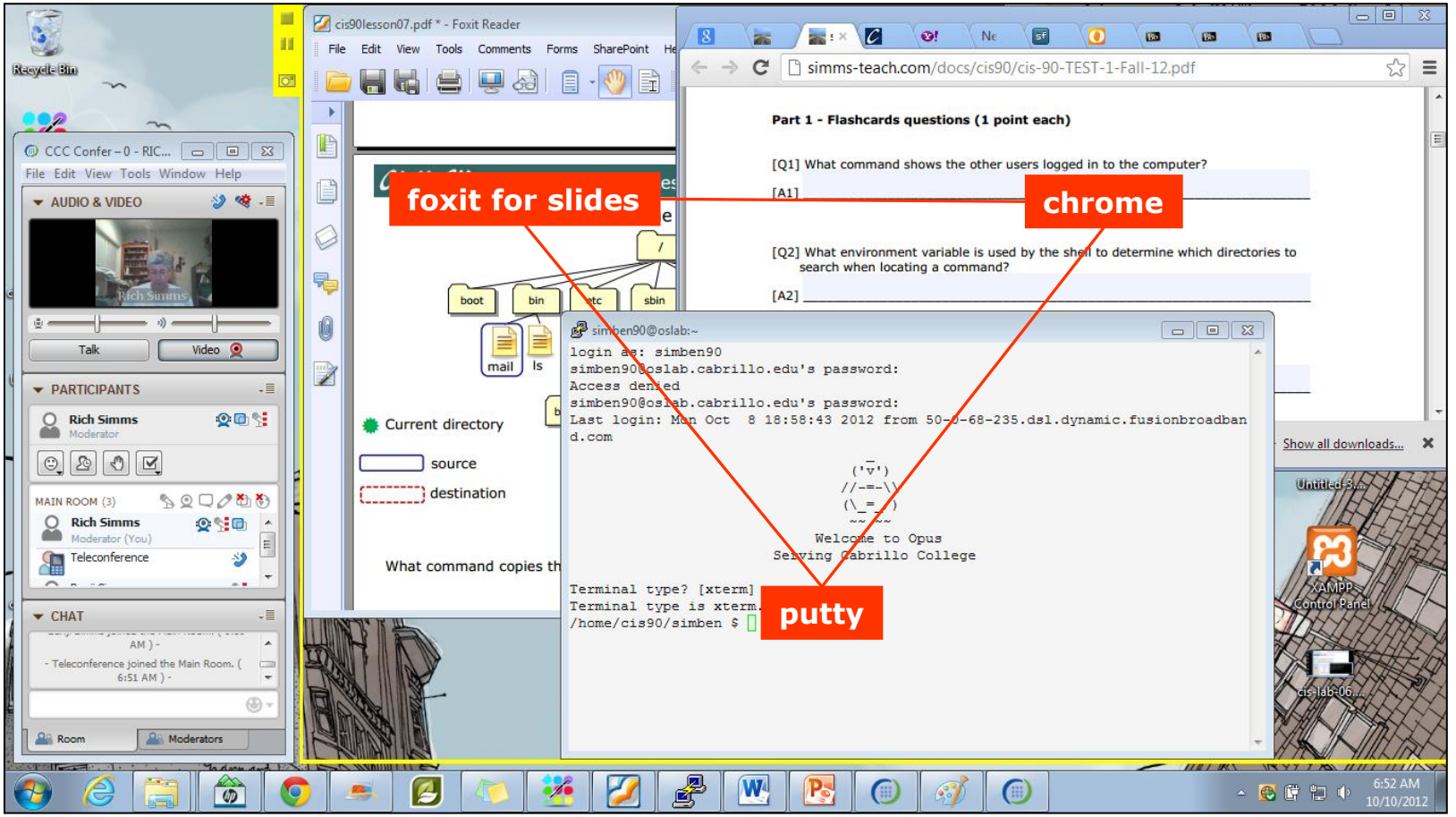


[ ] Toggle Talk button to not use Mic

CCC Confer

[ ] Video (webcam) optional

[ ] layout and share apps



Quiz

No Quiz Today !

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 12/5. 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 12/12.

4. Final Exam (Test #3) is Dec 12th 1-3:50PM in 2501 (no CCC Confer)

Code Name	Grading Choice	Quizzes & Tests													Forum			Labs										Project	Extra Credit	Total	Grade
		Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	T1	T2	T3	F1	F2	F3	F4	L1	L2	L3	L4	L5	L6	L7	L8	L9	L10				
Max Points		3	3	3	3	3	3	3	3	3	3	30	30	30	20	20	20	20	30	30	30	30	30	30	30	30	30	30	60	90	560	
anborn	grade	2	2	2		1	1	2	1	2	2	17	16		20	20	8		19	19	26		27	24	30	20	30	24		13		
arador	P/NP											19	9		4	4	20		26	29	0	17	28	27	29		23	24		9		
aragorn	grade		3	3	3	2	2	3	0	3	3	25	20		4	0	20		21	27	28	15	30		30	20	30	30		9		
balrog	grade	1	3		2					2		16	3		16	16	20		21	16	2	18	26	25		30	30			3		
bombadil	grade	3	2	3	3	3	3	3		3	3	28	27		0	20	20		28	21	30	25	29	30	29	30	29	28		15		
boromir	P/NP		3	3	3		3	2	3	2	3	23	16		20	16	20			28	2	22		22	30	30	28	29		12		
celeborn	grade	3	2	3	3	3	3	3	3	3	3	30	29		20	20	20		30	29	30	29	30	30	30	30	30	30		51		
dori	grade	3		3	3		0		3	3		5	12		4	16	16			2	20	13	27	24	27	20	24	25		9		
elrond	grade											12	11		20	20	20			26	30		27	22	30	30	30	30		9		
eomer	grade	3	3	3	3		2	2	2	3	3	26	23		0	0	4		27	29	28	23	29	30	29	30	30	30		15		
gimli	grade											14	2		0	0	0		17	26		10	26		30						
goldberry	P/NP	3	2						0			22	25		8	0	20		23	0	30		26	24	21	28	29	29		6		
huan	grade	3	3	3	3	3	1	3	0	3	3	28	26		20	20	12		28	30	30	28	30	30	30	30	30	30		45		
ingold	grade	3	3	3	3	3	3	3	3	3	3	28	16		20	20	20		30	27	30	26	30	30	30	30	28	30		13		
marhari	grade									2	2	25	17		0	20	8			0	30		30	27		30	30	28		36		
pallando	grade		1	3	2	3	3	3		3	3	13	11		20	16	20		22	21	30	7	24	26	28	30	14	22		15		
samwise	P/NP		3				2	3	3	3		24	11		8	12	8		21	27	26		30	30	29	28	30	30		8		
saruman	grade	3	3	3	3	3	3	3	3	3	3	28	29		20	20	20			30	30	29	30	30	30	30	30	30		25		
sauron	grade	1		0	3	3				3	3	27	27		20	20	20		29	30	30	30	29	30	30	30	30	30		42		
shadowfax	grade	3	3	3	3	3	2	3	3	3	3	29	29		20	20	20		30	30	30	29	30	30	30	30	26	30		25		
smeagol	grade	3	3	3	3	2	3	3	3	3	3	24	20		20	20	16		30	30	30	28	28	27	26	30	29	30		18		
theoden	grade	2	2	2	3	3	2	3	3	3	3	24	27		20	12	20		28	25	30	16	28	28	30	30	25	30		24		
tulkas	P/NP	0	2	3	3	0	3	3	3	3	3	23	17		20	20	20		0	26	28	22	24	30	27	30	29	30		10		

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 your progress using the Grades page of the course website

Tally as of 11/26/2012

anborn: 72% (328 of 450 points)
arador: 59% (268 of 450 points)
aragorn: 73% (331 of 450 points)
balrog: 55% (250 of 450 points)
bombadil: 92% (415 of 450 points)
boromir: 71% (320 of 450 points)
celeborn: 110% (497 of 450 points)
dori: 57% (259 of 450 points)
elrond: 70% (317 of 450 points)
eomer: 83% (377 of 450 points)
gimli: 27% (125 of 450 points)
goldberry: 65% (296 of 450 points)
huan: 104% (472 of 450 points)
ingold: 97% (438 of 450 points)
marhari: 63% (285 of 450 points)
pallando: 75% (340 of 450 points)
samwise: 74% (336 of 450 points)
saruman: 98% (441 of 450 points)
sauron: 110% (496 of 450 points)
shadowfax: 103% (467 of 450 points)
smeagol: 96% (435 of 450 points)
theoden: 94% (423 of 450 points)
tulkas: 84% (379 of 450 points)

Remaining point earning opportunities: T3=30, F4=20, Project=60 for 110 points plus up to 90 points maximum extra credit

Final Exam

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

It will be held in room 2501 on Wednesday, Dec 12th from 1:00 to 3:50PM (hard stop, no extension time period)

If you know you can't make this date you will need to contact the instructor, in advance, to arrange an exam **EARLIER** in the week.

No makeups after the Wednesday exam

Practice test will be available

	12/12	**Test #3 (the final exam)** **Time** • 1:00PM - 3:50PM in Room 2501 **Materials** • Presentation slides (download) • Test (download)		5 posts Lab X1 Lab X2

Make backup copies of your script

change change change change rest

```
/home/cis90/simben/bin $ cp myscript myscript.v1
```

change change change change rest

```
/home/cis90/simben/bin $ cp myscript myscript.v2
```

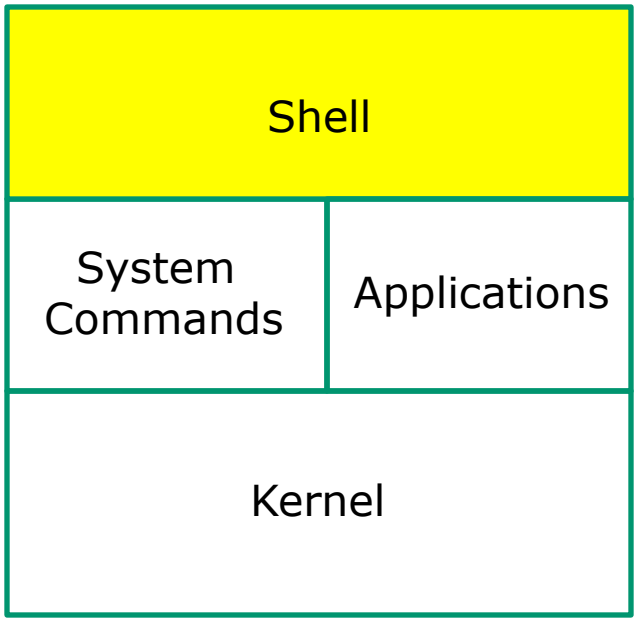
change change change change rest

```
/home/cis90/simben/bin $ cp myscript myscript.v3
```

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 Activity
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

rsimms@oslab:~

```
[rsimms@oslab ~]$ date
Tue Nov 27 08:05:20 PST 2012
[rsimms@oslab ~]$ /home/cis90/bin/checkmyscripts
Sean        -rwxr-xr-x. 1 calsea90 cis90 560 Nov 14 14:25 calsea/bin/myscript
Don         -rwxr-xr-x. 1 davdon90 cis90 760 Nov 21 16:16 davdon/bin/myscript
Carlile     -rwxr-x---. 1 ellcar90 cis90 481 Nov 23 20:25 ellcar/bin/myscript
Andrew      -rwxr-xr-x. 1 evaand90 cis90 1644 Nov 21 15:25 evaand/bin/myscript
Shahram     -rwxr-xr-x. 1 farsha90 cis90 769 Nov 21 15:35 farsha/bin/myscript
Carter      -rwxr-xr-x. 1 frocar90 cis90 557 Nov 14 14:23 frocar/bin/myscript
Sean        ls: cannot access fyosea/bin/myscript: No such file or directory
Dajan       -rwxr-xr-x. 1 hendaj90 cis90 1198 Nov 24 16:41 hendaj/bin/myscript
Bryn        -rwxr-xr-x. 1 kanbry90 cis90 521 Nov 21 15:18 kanbry/bin/myscript
Rita        -rwxrwxr-x. 1 kenrit90 cis90 610 Nov 24 17:32 kenrit/bin/myscript
Kelly       -rwxr-xr-x. 1 libkel90 cis90 882 Nov 21 15:48 libkel/bin/myscript
Ben         -rwxr-x---. 1 lyoben90 cis90 948 Nov 19 15:31 lyoben/bin/myscript
Ray         -rwxr-xr-x. 1 marray90 cis90 2171 Nov 26 03:20 marray/bin/myscript
Chad        ls: cannot access mescha/bin/myscript: No such file or directory
Michael     ls: cannot access mesmic/bin/myscript: No such file or directory
Homer       -rwxr-x--x. 1 milhom90 cis90 4298 Nov 21 09:32 milhom/bin/myscript
Evan        -rwxr-xr-x. 1 noreva90 cis90 551 Nov 14 14:21 noreva/bin/myscript
Josh        ls: cannot access potjos/bin/myscript: No such file or directory
Carlos      ls: cannot access ramcar/bin/myscript: Permission denied
Gustavo     -rwxr-x--x. 1 ramgus90 cis90 720 Nov 20 18:11 ramgus/bin/myscript
Jessica     ls: cannot access rawjes/bin/myscript: No such file or directory
Duke        ls: cannot access rodduk/bin/myscript: No such file or directory
Benji       -rwxr-xr-x. 1 simben90 cis90 10489 Nov 10 15:23 simben/bin/myscript
Evie        -rwxr-x---. 1 verevi90 cis90 594 Nov 14 14:36 verevi/bin/myscript
Jacob       ls: cannot access wiljac/bin/myscript: No such file or directory
Humberto    ls: cannot access zamhum/bin/myscript: Permission denied
[rsimms@oslab ~]$
```

*Instructor reminder: run **checkmyscripts** to see current status*

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 O.
        8) Gabriel
        9) George
        10) Glen
        11) Jaime
        12) Janet
        13) Joe F.
        14) Joe P.
        15) Junious
        16) Kang
        17) Lieven
        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/cis90/hussabil/bin/myscript
                        ;;
          2)    # Craig
                        /home/cis90/langlcra/bin/myscript
                        ;;
          3)    # Dan
                        /home/cis90/conydan/bin/myscript
                        ;;
          4)    # Doug
                        /home/cis90/kittldou/bin/myscript
                        ;;
          5)    # Duke
                        /home/cis90/roddyduk/bin/myscript
                        ;;
          6)    # Edgar D.
                        /home/cis90/delacedg/bin/myscript
                        ;;
          7)    # Edgar O.
                        /home/cis90/ortegedg/bin/myscript
                        ;;
          8)    # Gabriel
                        /home/cis90/pantogab/bin/myscript
                        ;;
          9)    # George
                        /home/cis90/balesgeo/bin/myscript
                        ;;
          10)   # Glen
                        /home/cis90/matligle/bin/myscript
                        ;;
          11)   # Jaime
                        /home/cis90/cervaja1/bin/myscript
                        ;;
          12)   # Janet
                        /home/cis90/tumajan/bin/myscript
                        ;;
          13)   # Joe F.
                        /home/cis90/ferrajoe/bin/myscript
                        ;;
          14)   # Joe P.
                        /home/cis90/pragejoe/bin/myscript
                        ;;
          15)   # Junious
                        /home/cis90/rosajun/bin/myscript
                        ;;
          16)   # Kang
                        /home/cis90/leekan/bin/myscript
                        ;;
          17)   # Lieven
                        /home/cis90/mambulie/bin/myscript
                        ;;
          18)   # Linda
                        /home/cis90/donohlin/bin/myscript
                        ;;
          19)   # Michael
                        /home/cis90/georgmic/bin/myscript
                        ;;
          20)   # Patrick
                        /home/cis90/caseypat/bin/myscript
                        ;;
          21)   # Talley
                        /home/cis90/senantal/bin/myscript
                        ;;
          22)   # Todd
                        /home/cis90/krametod/bin/myscript
                        ;;
          23)   # William
                        /home/cis90/tumawil/bin/myscript
                        ;;
          24)   # Benji
                        /home/cis90/simmsben/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
;;
*) 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
    clear
    echo -n "
        Spring 2009 cis 90 Projects
1) Bilal
2) Craig
3) Dan
4) Doug
5) Duke
6) Edgar O.
7) Edgar O.
8) Gabriel
9) George
10) Glen
    40) Songul
12) Janet
13) Joe F.
14) Joe P.
15) Junior
16) Kang
17) Lieven
18) Linda
19) Michael
20) Patrick
21) Talley
22) Todd
23) William
24) Wenji
99) Exit
    Enter Your Choice: "
    read RESPONSE
    case $RESPONSE in
        1)  # Bilal
            /home/cis90/husmabil/bin/myscript
            ;;
        2)  # Craig
            /home/cis90/langlcra/bin/myscript
            ;;
        3)  # Dan
            /home/cis90/coeydan/bin/myscript
            ;;
        4)  # Doug
            /home/cis90/kittdou/bin/myscript
            ;;
        5)  # Duke
            /home/cis90/coddyduk/bin/myscript
            ;;
        6)  # Edgar D.
            /home/cis90/delacedj/bin/myscript
            ;;
        7)  # Edgar O.
            /home/cis90/ortegedj/bin/myscript
            ;;
        8)  # Gabriel
            /home/cis90/pantogab/bin/myscript
            ;;
        9)  # George
            /home/cis90/haleageo/bin/myscript
            ;;
        10) # Glen
            /home/cis90/hatligle/bin/myscript
            ;;
        40)    # Songul
            /home/cis90/messison/bin/myscript
            ;;
        12) # Janet
            /home/cis90/tumajan/bin/myscript
            ;;
        13) # Joe F.
            /home/cis90/ferrajoe/bin/myscript
            ;;
        14) # Joe P.
            /home/cis90/pragejoe/bin/myscript
            ;;
        15) # Junior
            /home/cis90/corejun/bin/myscript
            ;;
        16) # Kang
            /home/cis90/leekan/bin/myscript
            ;;
        17) # Lieven
            /home/cis90/mambuli/bin/myscript
            ;;
        18) # Linda
            /home/cis90/donohlin/bin/myscript
            ;;
        19) # Michael
            /home/cis90/georgmic/bin/myscript
            ;;
        20) # Patrick
            /home/cis90/caseypat/bin/myscript
            ;;
        21) # Talley
            /home/cis90/essantal/bin/myscript
            ;;
        22) # Todd
            /home/cis90/kramertd/bin/myscript
            ;;
        23) # William
            /home/cis90/tumaswil/bin/myscript
            ;;
        24) # Wenji
            /home/cis90/simmswen/bin/myscript
            ;;
        99)    exit 0
            ;;
        *)  echo "Please enter a number between 1 and 4"
            ;;
    esac
    echo -e "Hit the Enter key to return to menu "
    read dummy
done
```

The while statement in allscripts will loop through the code forever

A case statement is used to run the appropriate myscript file in the student's bin directory. This is specified using an absolute filename.

`/home/cis90/messison/bin/myscript`

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

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 will loop forever. The only way out is the **exit** command in case 6)*

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
```

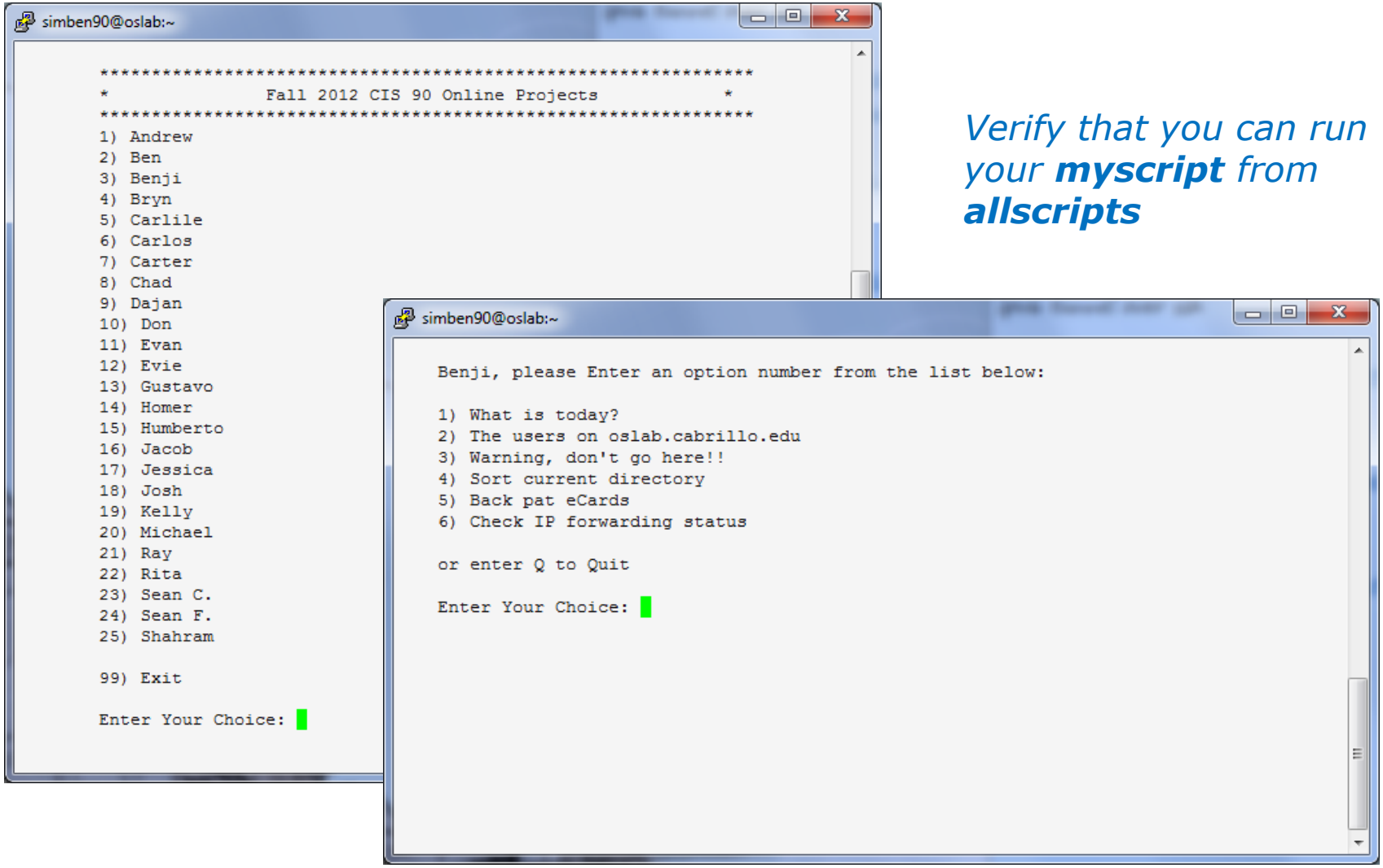
This is a case statement. One case for each task. Note the end of the case statement is case spelled backwards!

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 **read** command gets input from the user and stores it in a variable.*

*The variable to use is specified as the argument on the **read** command.*



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

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 one or more non-generic comments to explain what it is doing –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 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.
5	Present your script to 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 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)

This applies to each individual task

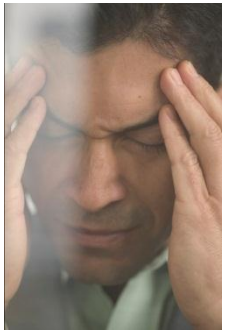
This applies to the project as a whole

Final Project

What takes longer?

Writing the script?

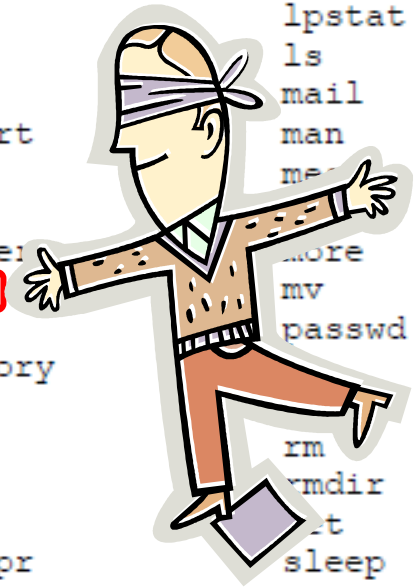
Or deciding what to script?



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

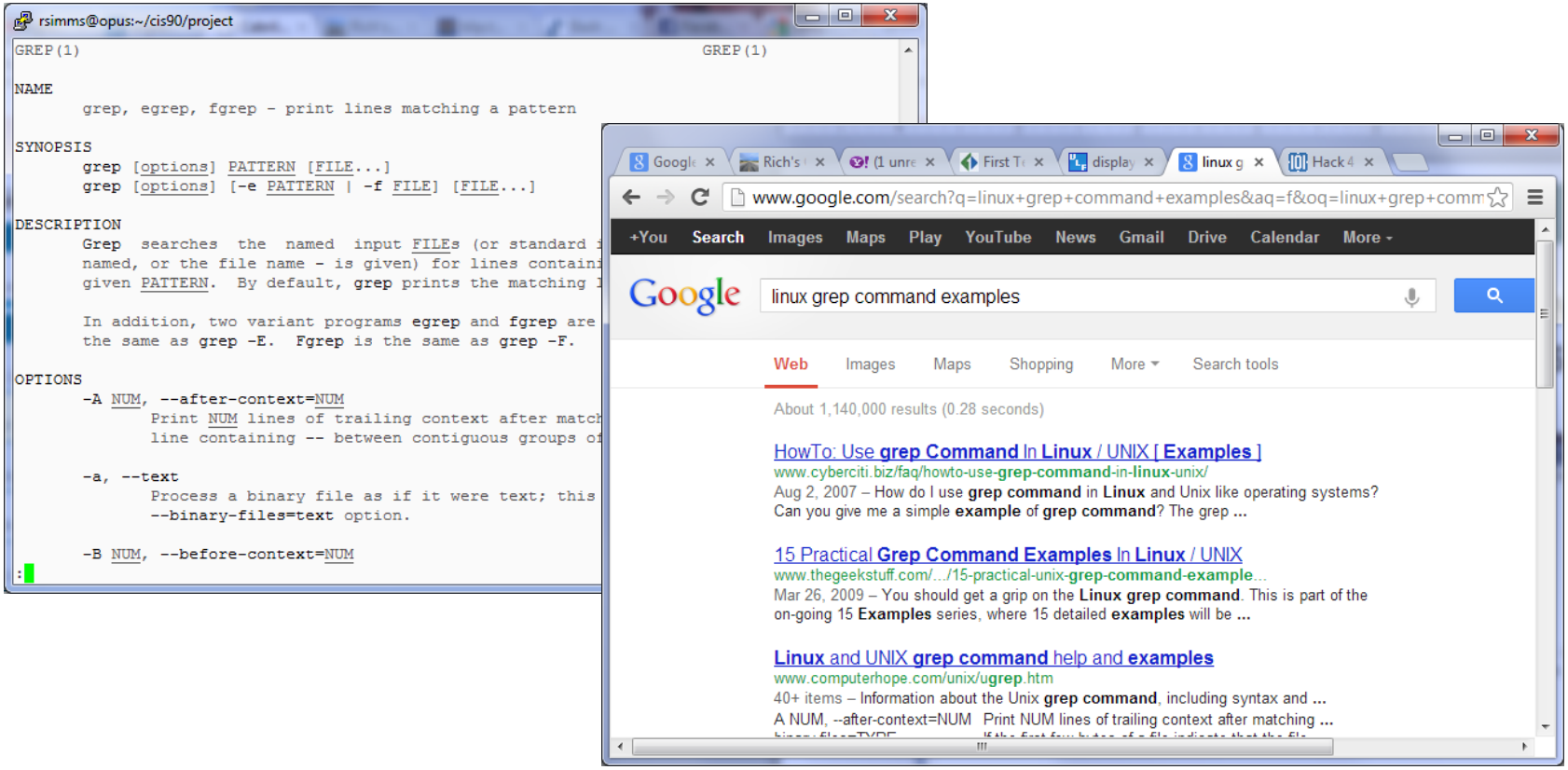
Commands

.	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		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



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

Before

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
                ;;
          2)    # Commands for Task
                ;;
"myscript" 37L, 546C
```

After

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
                ;;
-- INSERT --                                    10,5-12       Top
```

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*.

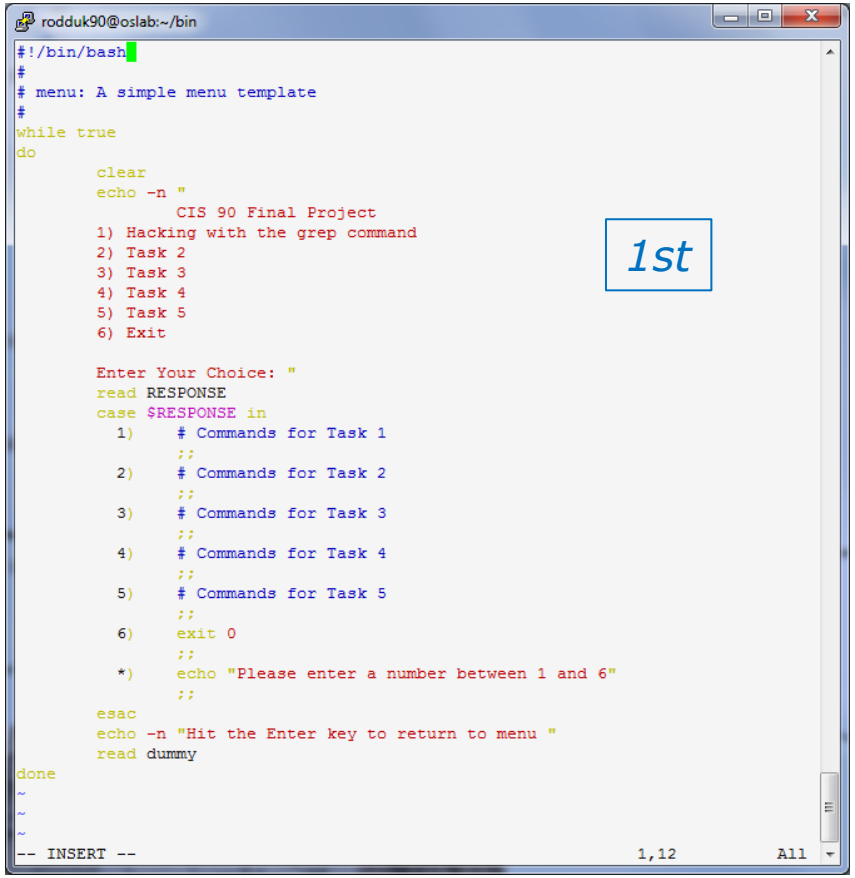
The term 'hacker' also tends to connote membership in the global community defined by the net (see *the network*. For discussion of some of the basics of this culture, see the How To Become A Hacker FAQ. It also implies that the person described is seen to subscribe to some version of the hacker ethic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

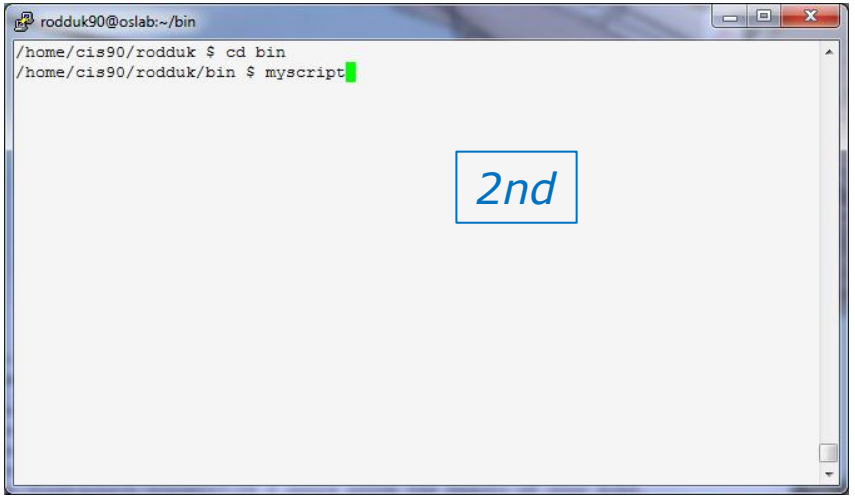
It is better to be described as a hacker by others than to describe oneself that way. Hackers consider themselves something of an elite (a meritocracy based on ability), though one to which new members are gladly welcome. There is thus a certain ego satisfaction to be had in identifying yourself as a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

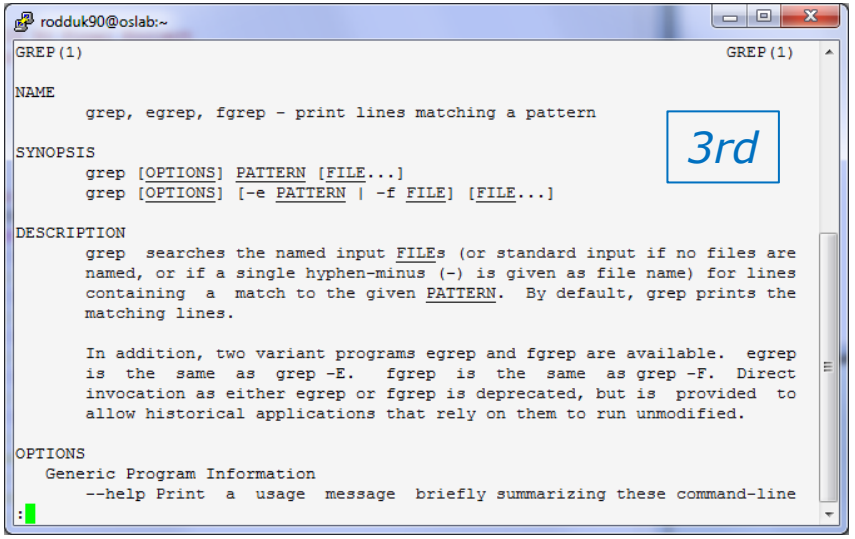
This term seems to have been first adopted as a badge in the 1960s by the hacker culture surrounding TMRC and the MIT AI Lab. We have a report that it was used in a sense close to this entry's by teenage radio hams and electronics tinkerers in the mid-1950s.

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

Layout your work area on the screen







Utilize screen real estate with multiple windows:

- *the 1st for vi,*
- *the 2nd for testing **myscript**,*
- *and a 3rd 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:
```

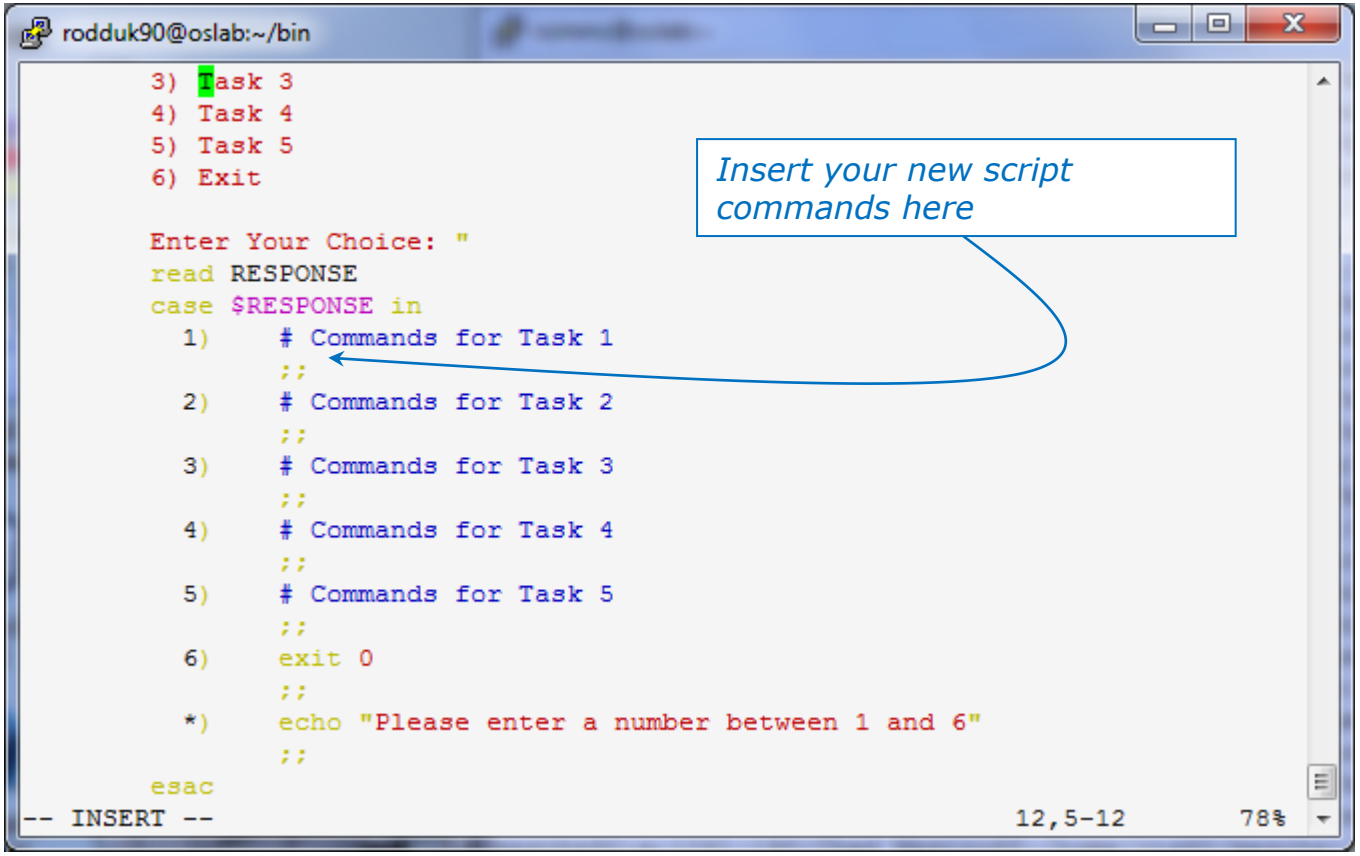
Changes work!

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 2nd window and verify your changes work*

Find the location to insert your new task commands



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
                ;;
          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" 38L, 593C written                                21,15-29      All
```

*Experiment with a **grep** command in 3rd window*

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

*Run **myscript** in the 2nd second window to test change.*

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 $
```

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
                ;;
          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" 38L, 612C written                                21,33-47      All
```

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

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

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!

```
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 $
```

Add some interaction

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

```
#!/bin/bash
#
# menu: A simple menu template
#
while true
do
        clear
        echo -n "
                CIS 90 Fin
        1) Hacking with th
        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
                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
                ;;
        esac
        echo -n "Hit the Enter key to return to menu "
        read dummy
done
"myscript" 40L, 711C written
```

rodduk90@oslab:~/bin

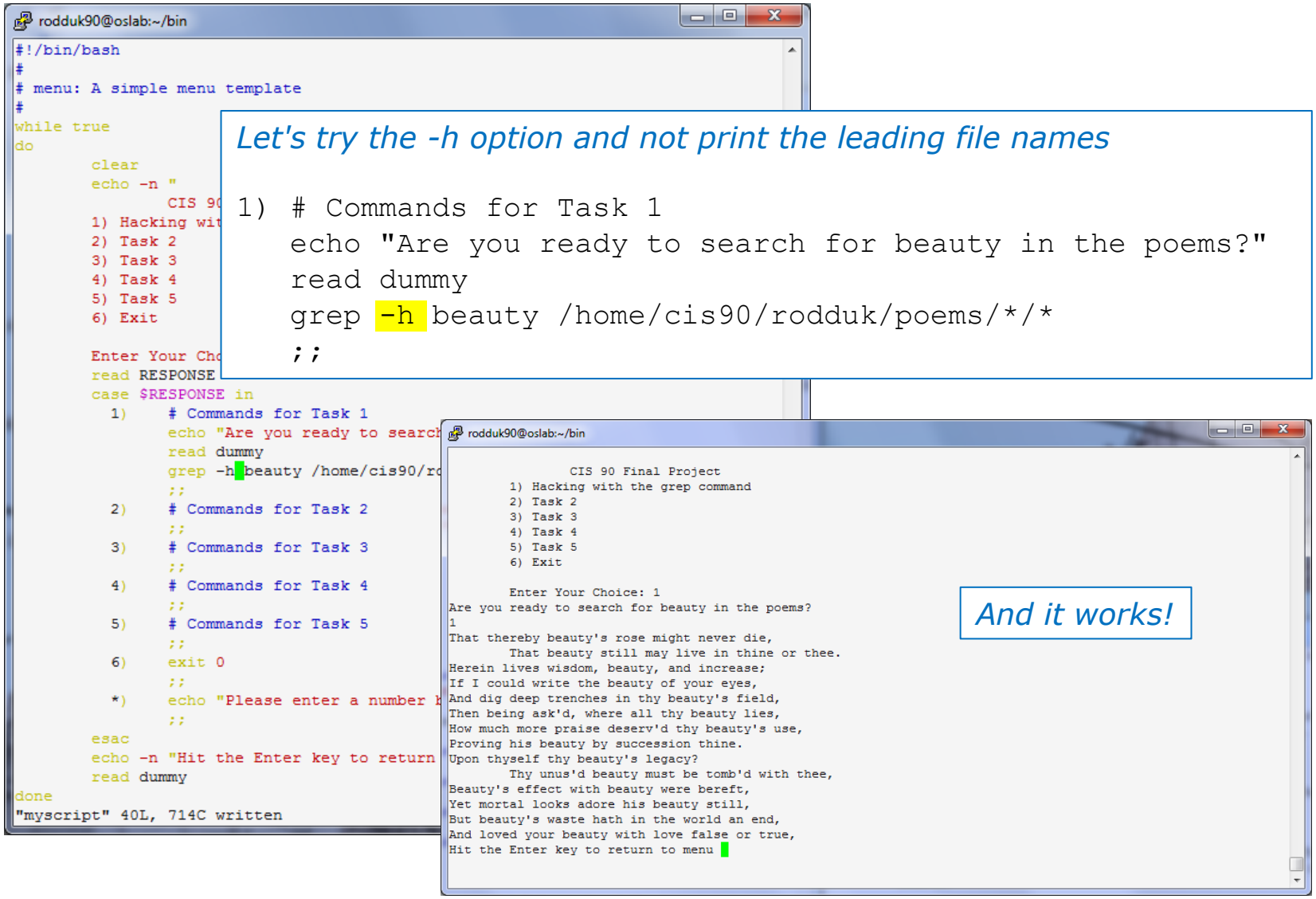
```
                CIS 90 Final Project
        1) Hacking with the grep command
        2) Task 2
        3) Task 3
        4) Task 4
        5) Task 5
        6) Exit

        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

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/*/*
    ;;
```



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
    ;;
```

```
#!/bin/bash
#
# menu: A s
#
while true
do
        cle
        ech

        1)
        2)
        3)
        4)
        5)
        6)

        Ent
        rea
        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
```

```
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?

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:

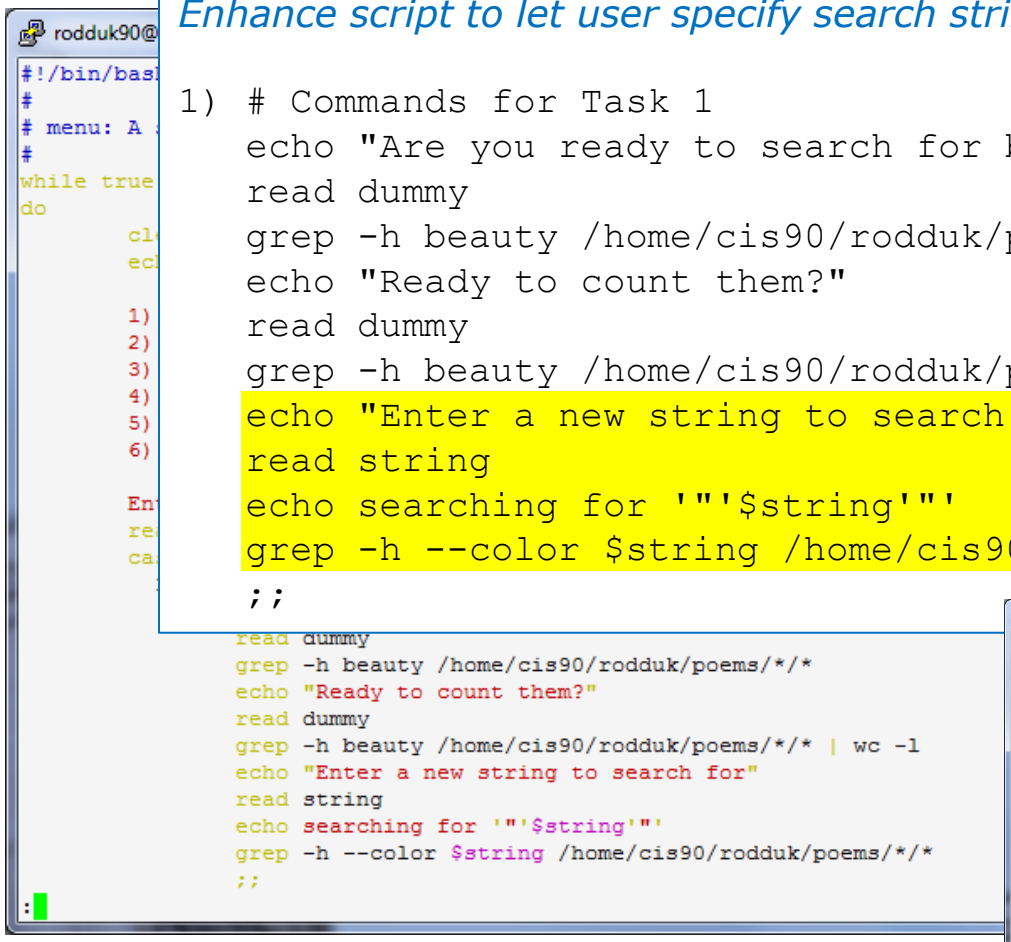
- Redirecting stdin (5 points)
- Redirecting stdout (5 points)
- Redirecting stderr (5 points)
- Use of permissions (5 points)
- ✔ Use of filename expansion characters (5 points)
- ✔ Use of absolute path (5 points)
- Use of relative path (5 points)
- Use of a PID (5 points)
- Use of inodes (5 points)
- 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)

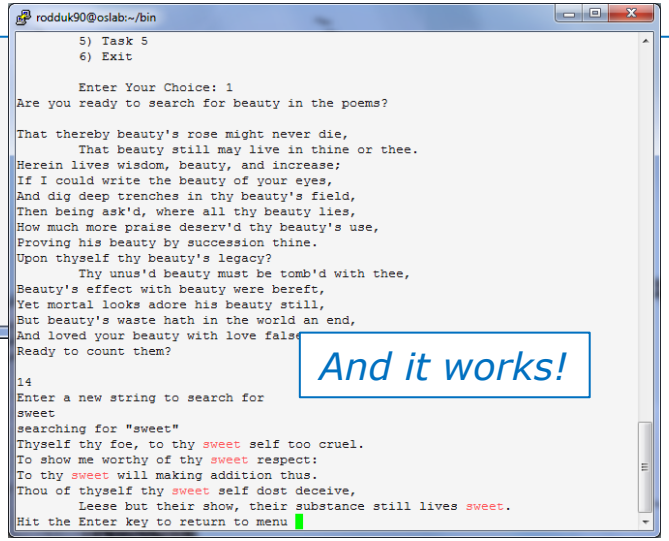
The maximum for this section is 25 points.

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
    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/*/*
    ;;
```





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/po
    echo "Enter a new string to search f
    read string
    echo searching for '"'$string'"'
    grep -h --color $string /home/cis90/
    ;;
```

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

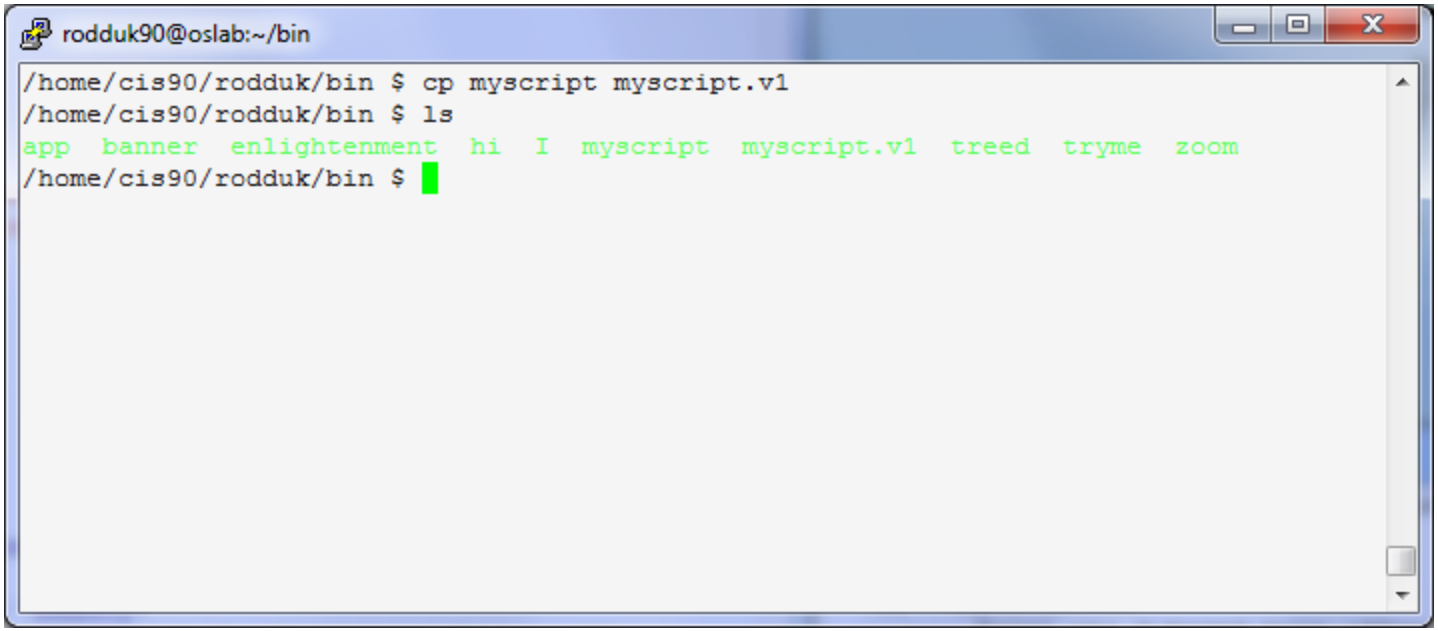
```
1)      # Task 1 - grep command explored

        # 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/poe
        ;;
```

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)

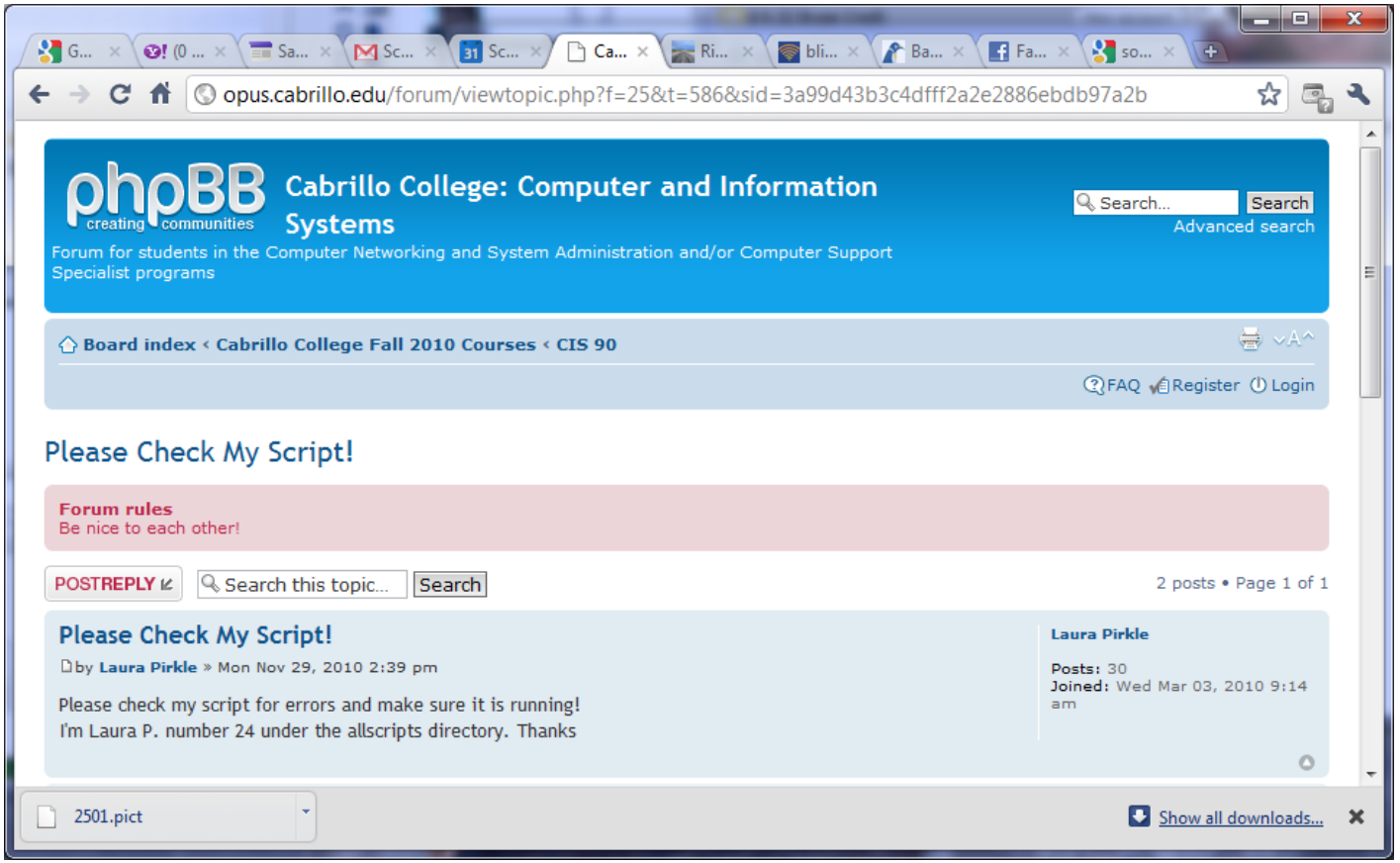
The maximum for this section is 25 points.

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



Make a backup copy of your hard work!

Testing your script



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

Plan extra time for:

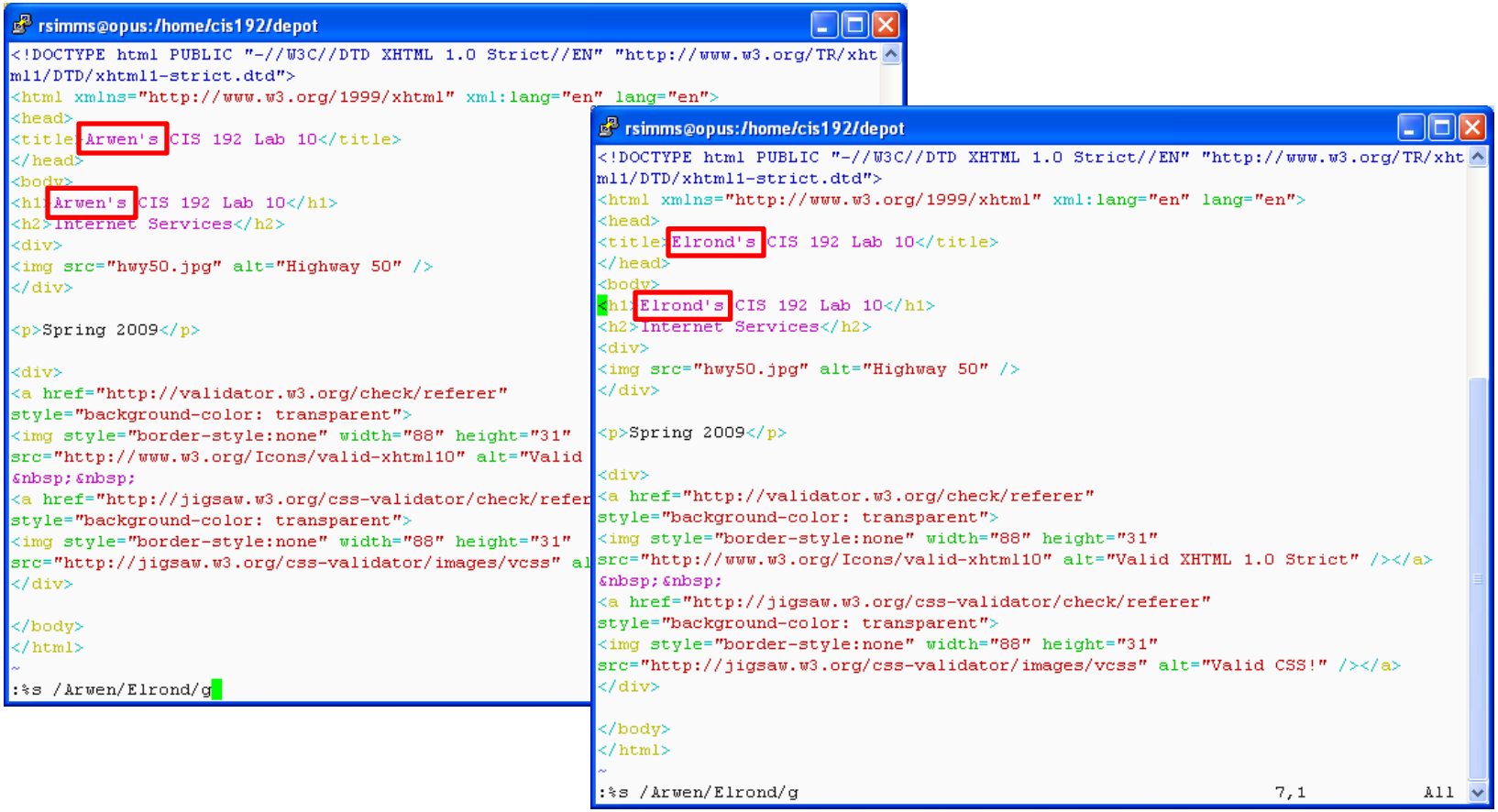
- Figuring our how to do what you really want to do!
- Removing syntax errors
- Removing logic errors
- Posting script code on the forum and asking others to view it and suggest how to fix it
- Sleeping on it

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

Scripting Tips
vi

Global search and replace with vi

esc:
%s /oldstring/newstring/g



Scripting Tips

$(cmd) or `cmd`

Shell Scripts

Sometimes you want to capture the output of a command and store in a variable or use as an argument

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 tics is an alternate way to do the same thing

Class Activity
Scripting

Try these commands:

```
date
banner date
banner `date`
banner $(date)
date | xargs banner
```

Which one will you use for Lab X1?

Scripting Tips

extracting a field from a record

/etc/passwd

```
[rsimms@opus ~]$ cat /etc/passwd
< snipped >
davdon90:x:1007:190:Don Davis:/home/cis90/davdon:/bin/bash
ellcar90:x:1008:190:Carlile Ellis:/home/cis90/ellcar:/bin/bash
frocar90:x:1009:190:Carter Frost:/home/cis90/frocar:/bin/bash
hendaj90:x:1010:190:Dajan Henk:/home/cis90/hendaj:/bin/bash
kanbry90:x:1011:190:Bryn Kanar:/home/cis90/kanbry:/bin/bash
kenrit90:x:1012:190:Rita Kennedy:/home/cis90/kenrit:/bin/bash
libkel90:x:1013:190:Kelly Libbey:/home/cis90/libkel:/bin/bash
lyoben90:x:1014:190:Ben Lyons:/home/cis90/lyoben:/bin/bash
marray90:x:1015:190:Ray Marr:/home/cis90/marray:/bin/bash
menfid90:x:1016:190:Fidel Mendoza:/home/cis90/menfid:/bin/bash
mesmic90:x:1017:190:Michael Messina:/home/cis90/mesmic:/bin/bash
noreva90:x:1018:190:Evan Norbom:/home/cis90/noreva:/bin/bash
potjos90:x:1023:190:Josh Potter:/home/cis90/potjos:/bin/bash
ramgus90:x:1024:190:Gustavo Ramirez:/home/cis90/ramgus:/bin/bash
< snipped >
```

*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
       ;;
```

Let's start with something simple like printing the current date and time

```
                Homer'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
        ;;
```

Let's add a friendly Hello using the users's logname

```
                Homer'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 milhom90
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 include the /etc/passwd info as well

```
                Homer'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 milhom90
milhom90:x:1156:103:Homer Miller:/home/cis90/milhom:/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
       ;;
```

*Cut the 5th field from the /etc/passwd record. The **-d** option specifies the delimiter to use.*

```
                Homer'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 milhom90
Homer Miller
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 and then use it

```
                Homer'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 milhom90
Hello Homer Miller
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

```
                Homer'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 Homer Miller

```
Wed Dec  3 14:07:07 PST 2008
Hit the Enter key to return to menu
```

myscript

```
8)      # Commands for Task 8
        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

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

        Enter Your Choice: 8
Hello Homer
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

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

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

        ;;
```

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

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

*An **if** statement is ended with **fi** (if spelled backward)*

myscript

```
          Homer'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

```
            Homer'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 14) Homer
- select Task 5 and enter d (for date)
- select Task 5 and enter c (for calendar)

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

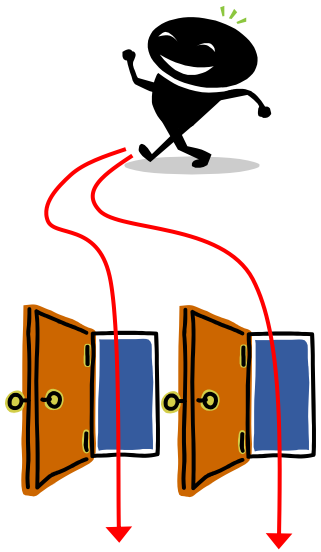
- **vi /home/cis90/milhom/bin/myscript**

Scripting Tips

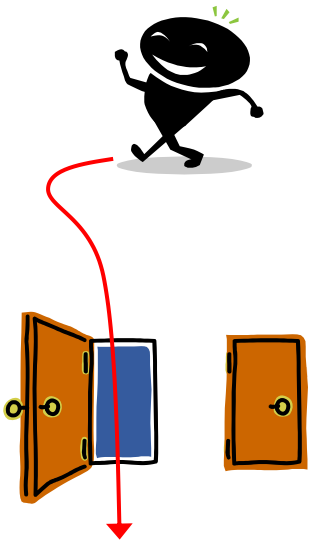
if statement with "or"

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

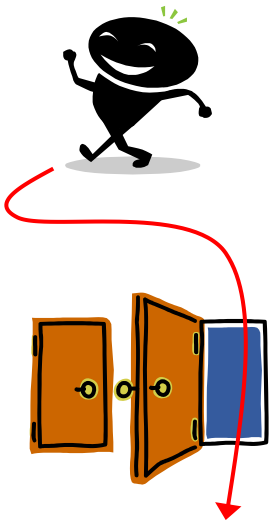
OR logic



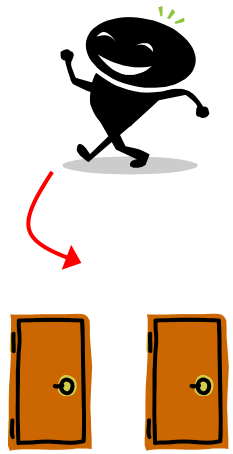
Yes



Yes



Yes



No

myscript

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

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

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

        ;;
```

*Run **date** if the user types d or D*

*Run **cal** if the user types c or C*

*The **||** is the logical "or" operator*

myscript

```
          Homer'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

```
          Homer'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 05:07:38 PDT 2009
 Hit the Enter key to return to menu
```

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

date *is run because user typed a D*

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
```

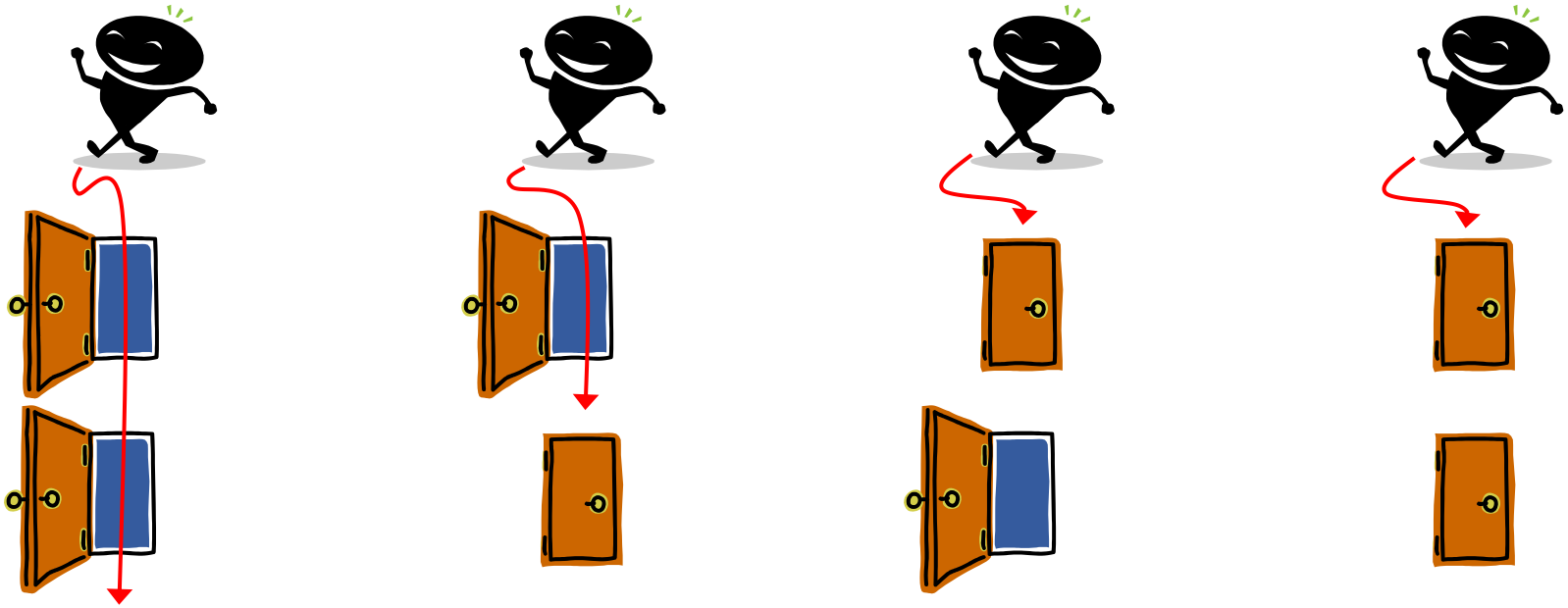
Prepare and run your script
chmod +x example654
example654

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

```
 Homer'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

```
                Homer'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 14) Homer
- select Task 7 several times with different answers

Now look at Homer's code to see how it was done:
- **vi /home/cis90/milhom/bin/myscript**

Scripting Tips

if

file types

myscript

```
4)      # More example IF statments
        echo "The files in this directory are: "
        ls -1
        echo -n "Which file are you interested in? : "
        read filename

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

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

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

tests to see if it's a regular file

tests to see if it's a directory

myscript

```
        Homer'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: 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

```
        Homer'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: 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 milhom90 cis90 4096 Apr  8 10:01 Anon
Hit the Enter key to return to menu
```

Additional file attributes to test for:

`-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.

Source: http://www.comptechdoc.org/os/linux/programming/script/linux_pgscripttest.html

Class Exercise

Run the previous example task
- run **allscripts**
- select 14) Homer
- select Task 4

Now look at Homer's code to see how it was done:
- **vi /home/cis90/milhom/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'
        ;;
```

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

        Enter Your Choice: 3
Hello Homer Miller
Wednesday
Wednesday, December 03, 2008
Hit the Enter key to return to menu
```

How can we do just one format or the other?

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
        ;;
```

Prompt user for choice then use if-then-else statement

```
Enter Your Choice: 3
Hello Homer Miller
Homer Miller, 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 Homer Miller
Homer Miller, 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.milhom90
myscript.v1 newscript old program quiet quiet.bak script treed tryme
typescript zoom

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

[rsimms@opus bin]$ echo $3
banner

[rsimms@opus bin]$ echo $7
I

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

[rsimms@opus bin]$ echo $#
20

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

A nice way to be able to reference specific files in a directory

```
[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
```

Another way to get a user's first name

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
```

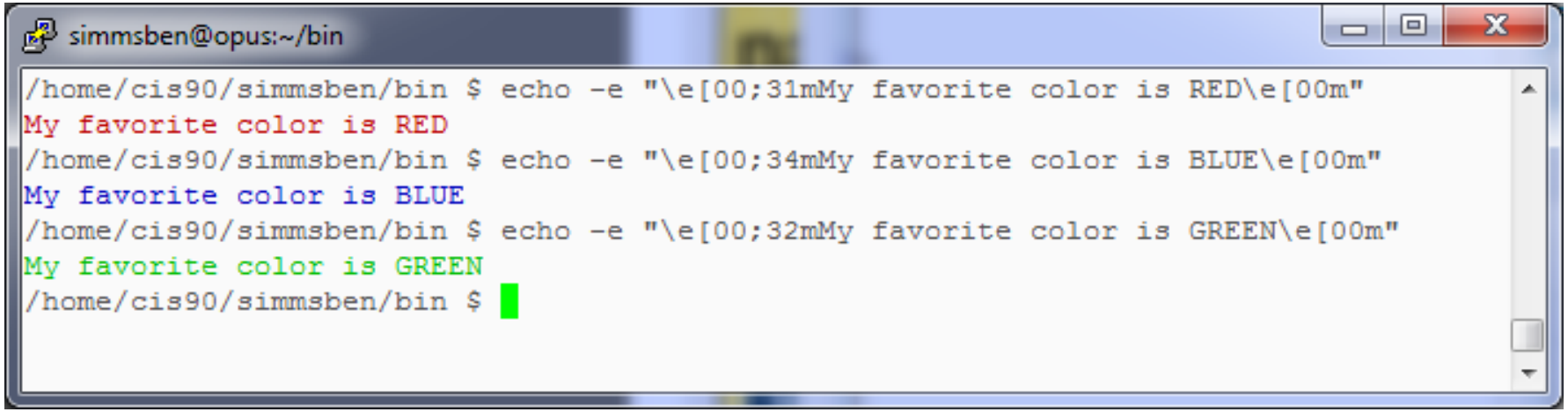
Prepare and run your script
chmod +x example777
example777

Scripting Tips

color

Using Color

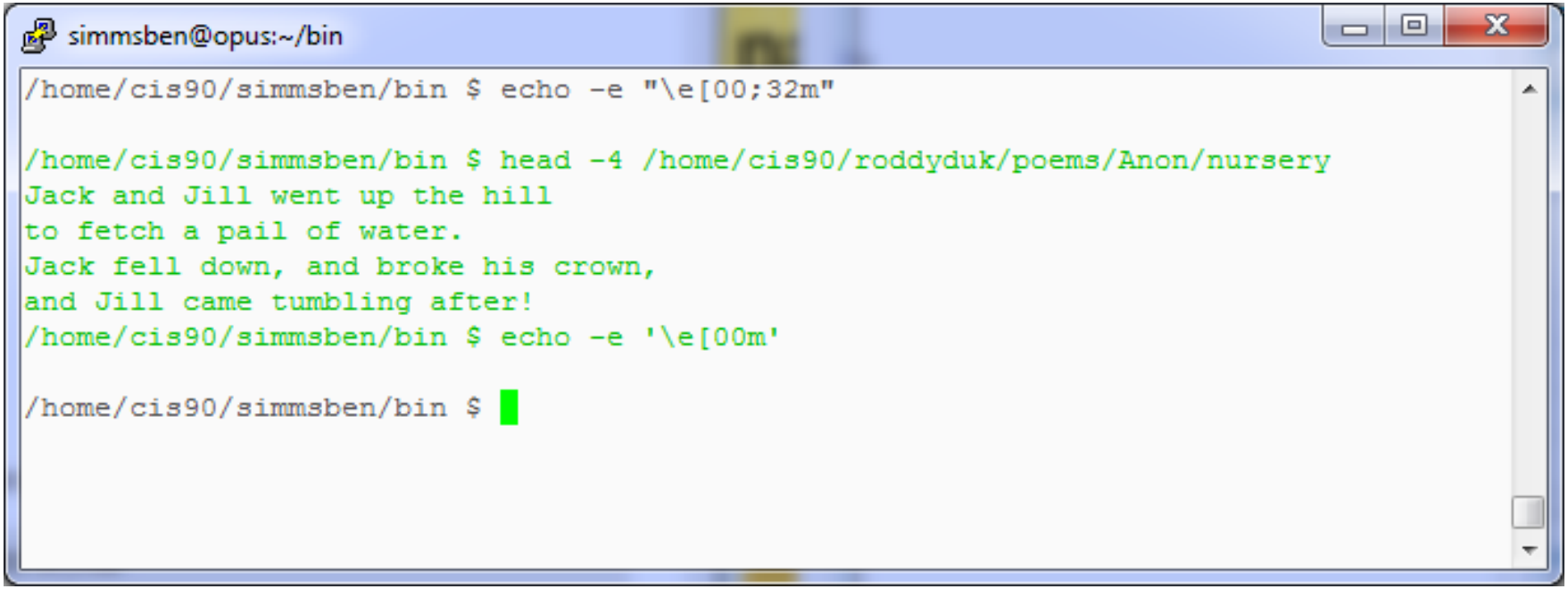
Black 0;30	Green 0;32	Red 0;31	Brown 0;33
Dark Gray 1;30	Light Green 1;32	Light Red 1;31	Yellow 1;33
Blue 0;34	Cyan 0;36	Purple 0;35	Light Gray 0;37
Light Blue 1;34	Light Cyan 1;36	Light Purple 1;35	White 1;37



*Use **echo -e "\e[0**n;nn**m"** to turn on color*
(the -e option enables interpretation of backslash escapes)

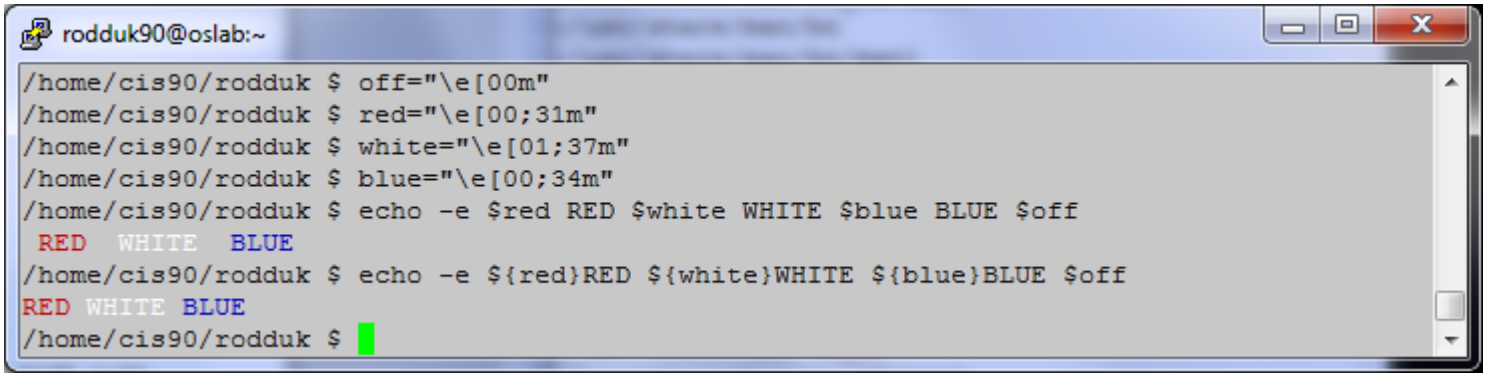
Source: http://hacktux.com/bash/colors

Using Color



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

Source: http://hacktux.com/bash/colors




```
/home/cis90/rodduk $ off="\e[00m"
/home/cis90/rodduk $ red="\e[00;31m"
/home/cis90/rodduk $ white="\e[01;37m"
/home/cis90/rodduk $ blue="\e[00;34m"
/home/cis90/rodduk $ echo -e $red RED $white WHITE $blue BLUE $off
 RED  WHITE  BLUE
/home/cis90/rodduk $ echo -e ${red}RED ${white}WHITE ${blue}BLUE $off
RED WHITE BLUE
```

Demonstrating the use of variables and curly braces to make color easier to use.

Curly braces are used to clearly delineate the variable name when there is no blank used as a separator from the next string

Scripting Tips

home directories and user names

Going from CIS 90 home directory name ➔ 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 ➔ 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 ➔ 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

Opus

One RHEL5 server per class

Internet

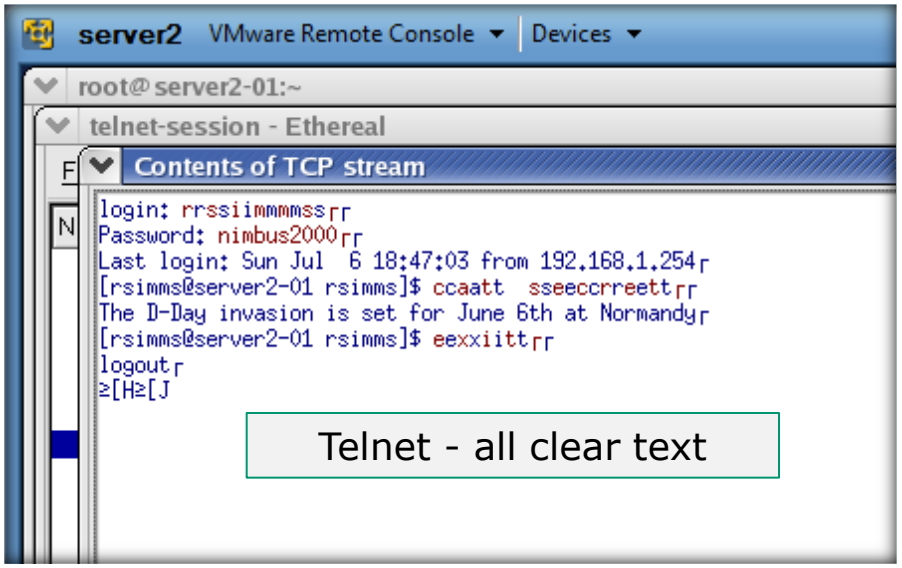
One RH9 Linux VM per laptop

One Windows laptop per student

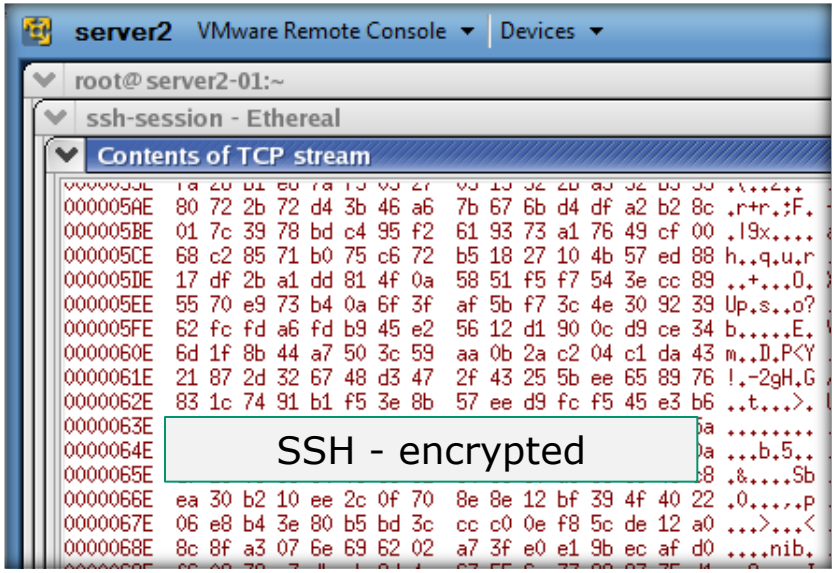
SSH is a network protocol that enables secure connections between computers

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 - for login and running remote commands
  - **scp** command - for copies files between systems

scp

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

- cp
  - copies files on the same system

```
cp /etc/hosts .
cp riddel 1 riddle2 riddles/
cp tally tally.v1
```

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

- scp
  - copies files between systems:

```
scp milhom90@oslab.cabrillo.edu:/etc/hosts .
scp riddle 1 riddle2 cis90@P1-Hugo:riddles/
scp -P 425 rsimms@frodo.simms-teach.com:tally tally.v1
```

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

The @ and : separators are always required with scp

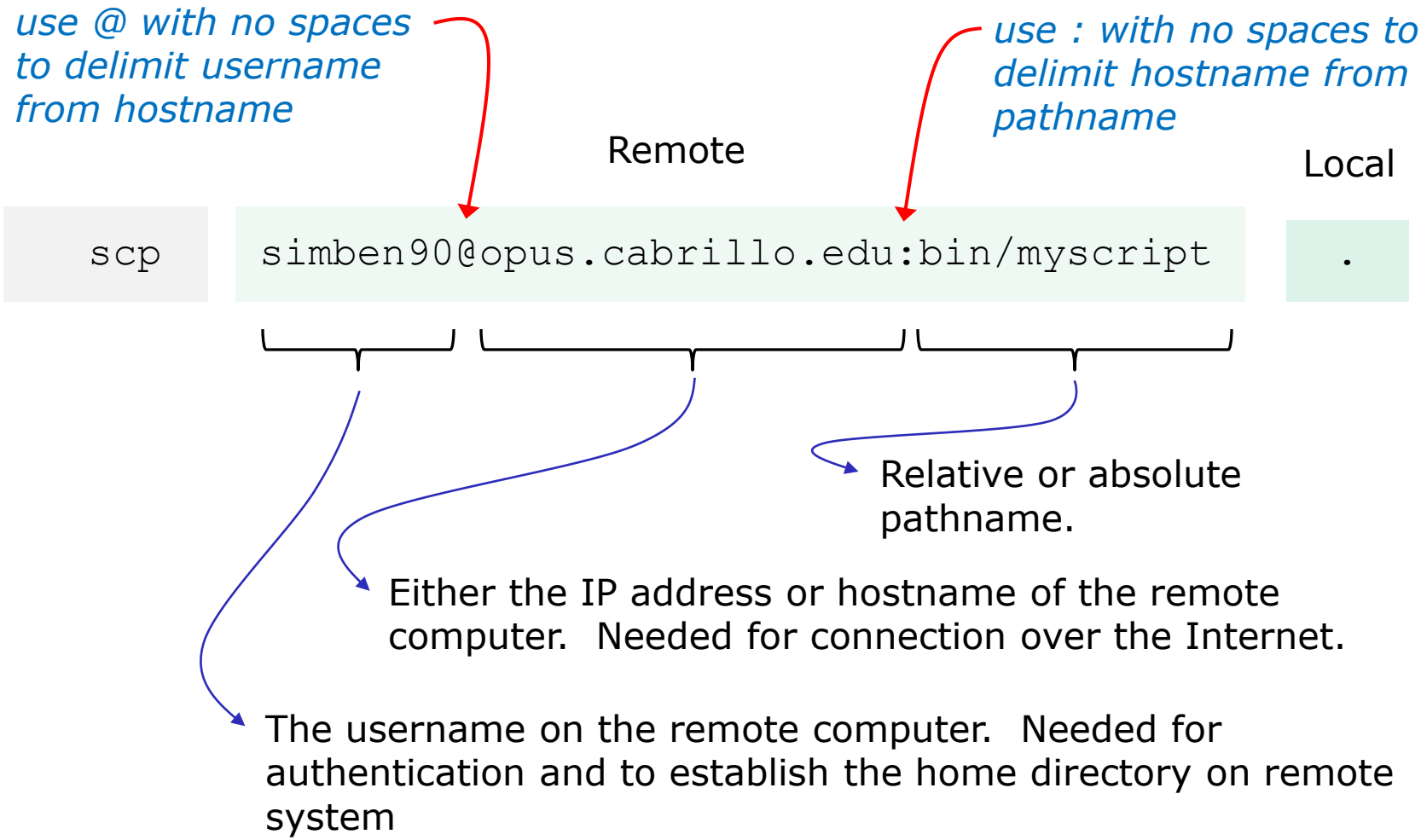
scp

	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



Copying a file from Opus to Sun-Hwa
(initiated from Sun-Hwa)

On Opus

```
/home/cis90/simben $ head -n1 ../depot/scrooge
Stave 2: The First of the Three Spirits
/home/cis90/simben $
```

On Sun-Hwa

```
[CISLAB\simben90@sun-hwa ~]$ head -n1 scrooge
head: cannot open `scrooge' for reading: No such file or directory

[CISLAB\simben90@sun-hwa ~]$ scp simben90@opus.cabrillo.edu:../depot/scrooge .
simben90@opus.cabrillo.edu's password:
scrooge                                       100%   33KB  33.1KB/s   00:00
[CISLAB\simben90@sun-hwa ~]$

[CISLAB\simben90@sun-hwa ~]$ head -n1 scrooge
Stave 2: The First of the Three Spirits
[CISLAB\simben90@sun-hwa ~]$
```

Copying multiple files from Opus to Sun-Hwa
(initiated from Sun-Hwa)

On Opus

```
/home/cis90/simben $ ls bin
app      datecal         hi  myscript     myscript.v2  simple.c  tryme
banner   enlightenment   I   myscript.v1  simple       treed     zoom
/home/cis90/simben $
```

On Sun-Hwa

```
[CISLAB\simben90@sun-hwa ~]$ ls bin
ls: cannot access bin: No such file or directory
[CISLAB\simben90@sun-hwa ~]$ mkdir bin
[CISLAB\simben90@sun-hwa ~]$ scp simben90@opus:bin/my* bin/
simben90@opus's password:
myscript                                      100%   10KB  10.2KB/s   00:00
myscript.v1                                   100%   10KB  10.2KB/s   00:00
myscript.v2                                   100%   10KB  10.2KB/s   00:00
[CISLAB\simben90@sun-hwa ~]$
```

Copying a file from Sun-Hwa to Opus
(initiated from Sun-Hwa)

On Opus

```
/home/cis90/simben $ ls file25
ls: cannot access file25: No such file or directory
```

On Sun-Hwa

```
[CISLAB\simben90@sun-hwa ~]$ echo "I love Linux" > file25
[CISLAB\simben90@sun-hwa ~]$ scp file25 simben90@opus:
The authenticity of host 'opus (172.30.5.20)' can't be established.
RSA key fingerprint is 7d:32:80:b9:52:32:c8:dc:3b:16:0e:ba:8c:fd:79:ef.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'opus,172.30.5.20' (RSA) to the list of known hosts.
simben90@opus's password:
file25                                        100%   13     0.0KB/s   00:00
[CISLAB\simben90@sun-hwa ~]$
```

```
/home/cis90/simben $ cat file25
I love Linux
```

Copying a file from Sun-Hwa to Opus
(initiated from Opus)

On Sun-Hwa

```
[CISLAB\simben90@sun-hwa ~]$ echo "I love dogs" > file15
[CISLAB\simben90@sun-hwa ~]$
```

On Opus

```
/home/cis90/simben $ cat file15
cat: file15: No such file or directory
/home/cis90/simben $

/home/cis90/simben $ scp cislab\\simben90@sun-hwa:file15 .
cislab\simben90@sun-hwa's password:
file15                                        100%   12     0.0KB/s   00:00
/home/cis90/simben $

/home/cis90/simben $ cat file15
I love dogs
/home/cis90/simben $
```

Copying a file from Sun-Hwa to Opus and renaming it
(initiated from Sun-Hwa)

On Opus

```
/home/cis90/simben $ cat iloveunix
cat: iloveunix: No such file or directory
/home/cis90/simben $
```

On Sun-Hwa

```
[CISLAB\simben90@sun-hwa ~]$ echo "I love UNIX" > file35
[CISLAB\simben90@sun-hwa ~]$ scp file35 simben90@opus:iloveunix
simben90@opus's password:
file35                                        100%   12     0.0KB/s   00:00
[CISLAB\simben90@sun-hwa ~]$
```

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

Class Activity

- On Opus, locate the *ptest.template* file in the CIS 90 *depot* directory
- Log into Sun-Hwa
- Copy the *ptest03.template* file in the CIS 90 *depot* directory to your home directory naming it *ptest03* at the same time

tar

tar command

tar *options tarfile files*

To simplify file transfers, Windows users typically "zip" multiple files together into a singe "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 misc directory

```
/home/cis90/simben $ ls misc
file.dos  fruit  manpage  mystery  salad  tiurf
what_am_i
/home/cis90/simben $
```

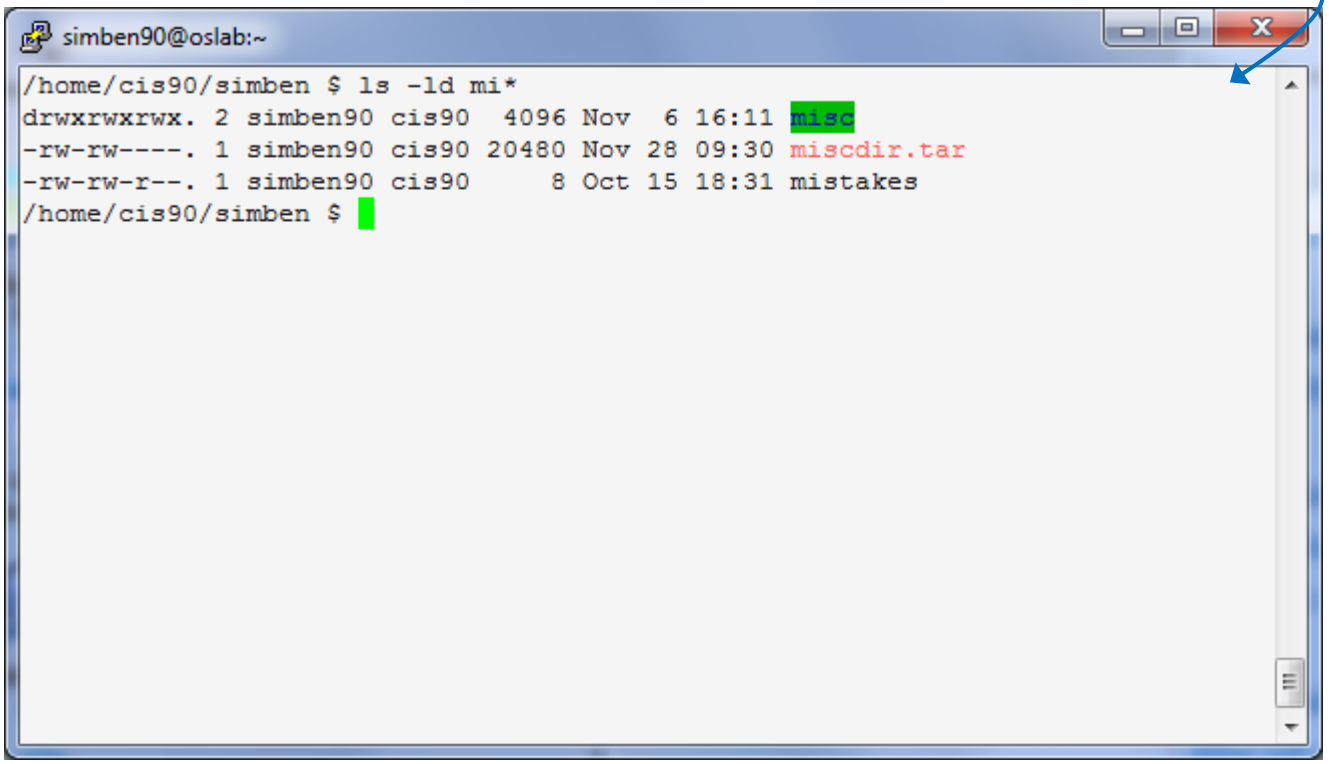
```
/home/cis90/simben $ tar cvf miscdir.tar misc/
misc/
misc/fruit
misc/file.dos
misc/salad
misc/mystery
misc/what_am_i
misc/manpage
misc/tiurf
/home/cis90/simben $
```

relative pathname to directory to archive

name of archive file

tar command

Tarballs show as red in listings



tar command

View contents of a tarball

```
/home/cis90/simben $ tar tvf miscdir.tar
drwxrwxrwx simben90/cis90    0 2012-11-06 16:11 misc/
-rw-r--r-- simben90/cis90   78 2004-10-26 16:36 misc/fruit
-rw-r--r-- simben90/cis90  148 2001-07-20 22:54 misc/file.dos
-rw-r--r-- simben90/cis90   78 2004-04-17 12:13 misc/salad
lrwxrwxrwx simben90/cis90    0 2012-08-01 16:55 misc/mystery -> ../bin/enlightenment
-rw-r--r-- simben90/cis90  352 2001-07-20 15:04 misc/what_am_i
-rw-r--r-- simben90/cis90 10576 2001-07-20 20:58 misc/manpage
-rw-rw-r-- simben90/cis90    78 2012-10-15 09:25 misc/tiurf
/home/cis90/simben $
```

tar command

On another Linux system (Sun-Hwa in VLab)

```
[CISLAB\simben90@sun-hwa ~]$ ls misc
ls: cannot access misc: No such file or directory
[CISLAB\simben90@sun-hwa ~]$
```

tar command

On another Linux system (Sun in Vlab)

```
[CISLAB\simben90@sun-hwa ~]$ ls misc
ls: cannot access misc: No such file or directory


[CISLAB\simben90@sun-hwa ~]$ scp simben90@opus:miscdir.tar .
simben90@opus's password:
miscdir.tar                                   100%   20KB  20.0KB/s   00:00
[CISLAB\simben90@sun-hwa ~]$


[CISLAB\simben90@sun-hwa ~]$ tar xvf miscdir.tar
misc/
misc/fruit
misc/file.dos
misc/salad
misc/mystery
misc/what_am_i
misc/manpage
misc/tiurf
[CISLAB\simben90@sun-hwa ~]$
```

Copy tarball from Opus

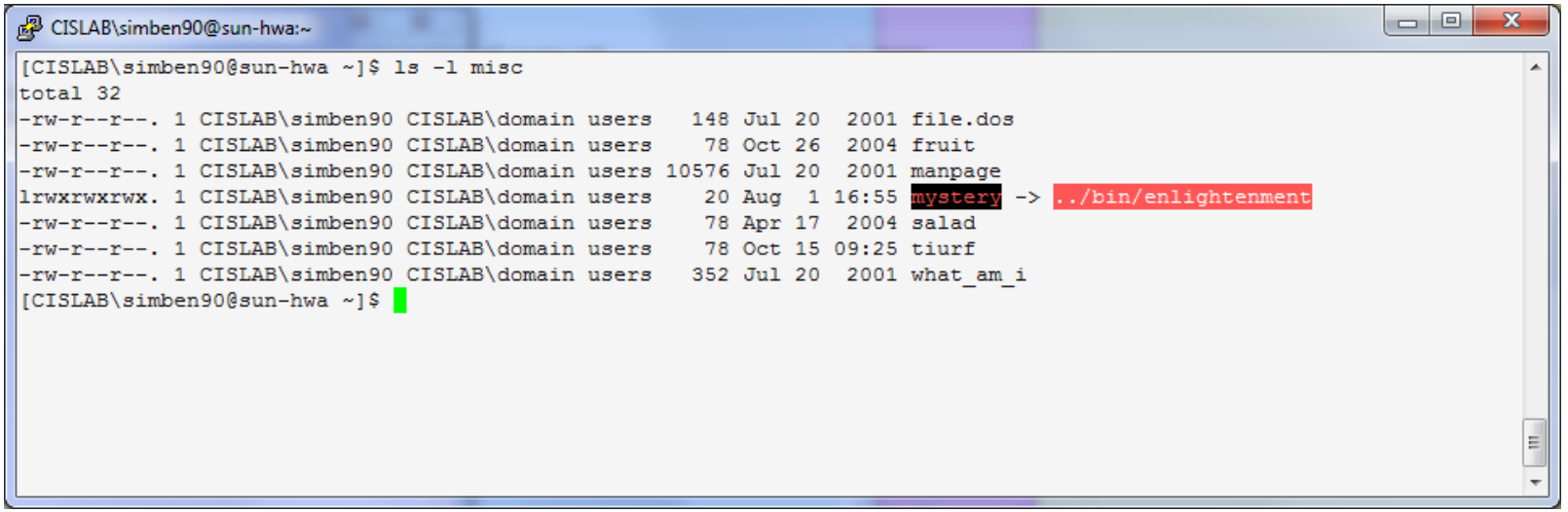
Extract tarball on Sun-Hwa

Note, misc/ directory is created and populated

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

tar command

After extraction



Note: the symbolic link is broken because there is no enlightenment file in local bin directory onSun-Hwa

Class Activity

Only do this if you have not started Lab X2 already:

- On Opus, locate the *dogs.tar* tarball in the CIS 90 *depot* directory
- Copy it to your home directory
- Extract the contents to your home directory
- List your new *dogs/* directory

Wrap up

Commands:

basename	- extract filename form pathname
scp	- secure copy command
tar	- archive command
if then else	- conditionals in scripts
[ ]	- for logic tests in scripts

Next Class

Project is due next week!

Backup