

Oregon Institute of Technology

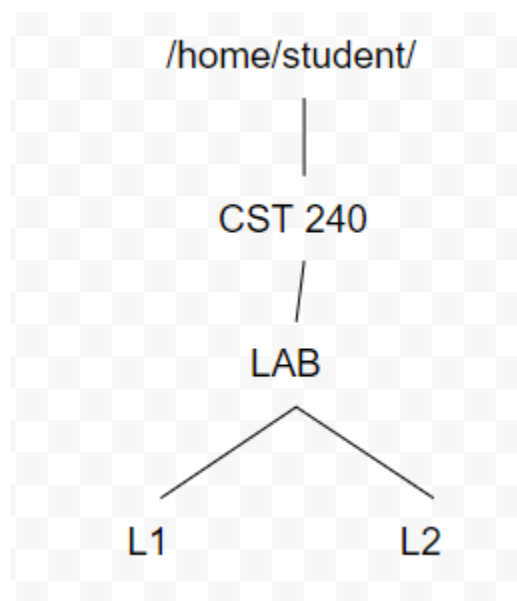
CST 240 Linux Programming Lab #1: Bash, C Review, Primes

CST 240

LAB1: Bash, C Review, Primes

Part 1: Bash

1.(10 pts) Create the directory structure below in your own Unix directory. Show (with a copy and paste) the results of appropriate commands in the space below to show that you made the correct directories.



i.e. show results of commands below in this order

```
$cd ~
```

```
$pwd
```

```
$ls
```

```
$ls CST240
```

```
$ls CST240/LAB
```

```
cari@Thinky:~$ ls
```

```
Desktop  Documents  Downloads  Music  Pictures  Public  
Templates  Videos
```

```
cari@Thinky:~$ mkdir CST240
```

```
cari@Thinky:~$ ls
```

```
CST240  Documents  Music  Public  Videos  
Desktop  Downloads  Pictures  Templates
```

```
cari@Thinky:~$ cd CST240
```

--	--	--

Oregon Institute of Technology

CST 240 Linux Programming Lab #1: Bash, C Review, Primes

```
cari@Thinky:~/CST240$ mkdir LAB
cari@Thinky:~/CST240$ ls
LAB
cari@Thinky:~/CST240$ cd LAB
cari@Thinky:~/CST240/LAB$ mkdir L1
cari@Thinky:~/CST240/LAB$ mkdir L2
cari@Thinky:~/CST240/LAB$ ls
L1  L2
cari@Thinky:~/CST240/LAB$ cd ~
cari@Thinky:~$ pwd
/home/cari
cari@Thinky:~$ ls
CST240  Documents  Music      Public     Videos
Desktop Downloads  Pictures   Templates
cari@Thinky:~$ ls CST240
LAB
cari@Thinky:~$ ls CST240/LAB
L1  L2
```

2. (10 Pts) In the directory L1, create a file named "RichesIHold.txt". The content of the file is shown below.

Riches I hold in light esteem
And Love I laugh to scorn
And lust of Fame was but a dream
That vanished with the morn—

And if I pray, the only prayer
That moves my lips for me
Is—"Leave the heart that now I bear
And give me liberty."

Yes, as my swift days near their goal
'Tis all that I implore
Through life and death, a chainless soul
With courage to endure!

Use the cat command to show RichesIHold.txt above. Copy and paste the cat command and content of the file in the space below.

```
cari@Thinky:~/CST240/LAB/L1$ cat RichesIHold.txt
```

--	--	--

	<h2>Oregon Institute of Technology</h2>	
--	---	--

CST 240 Linux Programming
Lab #1: Bash, C Review, Primes

Riches I hold in light esteem
And Love I laugh to scorn
And lust of Fame was but a dream
That vanished with the morn—

And if I pray, the only prayer
That moves my lips for me
Is—"Leave the heart that now I bear
And give me liberty."

Yes, as my swift days near their goal
'Tis all that I implore
Through life and death, a chainless soul
With courage to endure!

Make a copy of the file above and change the name to TechIHold.txt. Use vi or emacs (or a Linux editor of your choice) to change TechIHold.txt in the following ways:

- Delete second paragraph ("And if I pray...").
- Copy the first paragraph ("Riches I hold") and paste this paragraph at the end of the file. After you have completed this step, you will have 2 copies of the paragraph starting with "Riches I hold".
- Change "Riches" to "Tech" and "courage" to "computers" throughout the poem.

After completing these changes in an editor, cat TechIHold.txt again. Copy and paste the cat command and content of TechIHold.txt in space below

```
cari@Thinky:~/CST240/LAB/L1$ vi TechIHold.txt
cari@Thinky:~/CST240/LAB/L1$ cat TechIHold.txt
Tech I hold in light esteem
And Love I laugh to scorn
And lust of Fame was but a dream
```

```
Yes, as my swift days near their goal
'Tis all that I implore
Through life and death, a chainless soul
With computers to endure!
```

--	--	--

	<h2>Oregon Institute of Technology</h2>	
--	---	--

	<h3>CST 240 Linux Programming Lab #1: Bash, C Review, Primes</h3>	
--	---	--

Tech I hold in light esteem
And Love I laugh to scorn
And lust of Fame was but a dream

3. (15 pts) This question explores some usage of braces, brackets and parentheses. Best to Google to learn more. But for now, use Unix to answer questions below
NOTE: YOU MUST HAVE THE EXACT SPACING AS SHOWN BELOW. IF NOT, ERROR MESSAGES WILL SHOW. IF CORRECT, THERE WILL BE A CORRECT RESULT FOR EVERY QUESTIONS. ON YOUR OWN, TRY INCORRECT SPACING AND SEE ERRORS

a. Type in:

```
$ VARIABLE=abcdef  
$ if [ $VARIABLE == abcdef ] ; then echo yes ; else echo no ; fi
```

What is the result after return? Why? it returns yes, because the VARIABLE, indeed, was abcdef, and we were testing for truth

b. Type in:

```
$ VARIABLE=abcdef  
$ if [[ $VARIABLE == abcdef ]] ; then echo yes ; else echo no ;  
fi
```

What is the result after return? Why? It returned yes, because the VARIABLE, indeed, was abcdef, and we were testing for true/false

c. Type in:

```
$ VARIABLE=abcdef  
$ if [ $VARIABLE == 123456 ] ; then echo yes ; else echo no ; fi
```

What is the result after return? Why? it returns no, because the variable is abcdef, not 123456, and we were testing for truth

d.

```
$ pwd  
/home/student  
$ (cd /tmp; pwd)
```

--	--	--

	<h2>Oregon Institute of Technology</h2>	
--	---	--

	<h3>CST 240 Linux Programming Lab #1: Bash, C Review, Primes</h3>	
--	---	--

What is the result after return? Why? the return is /tmp, because the () creates a subshell, meaning that a /tmp directory was made and then we printed the working directory in that subshell

Now type:

```
$ pwd
```

What is the result after return? Why? /home/cari, because the subshell is only temporary, and its “scope”, as it were, finished at the end of the parentheses. Now out of that scope, /tmp no longer exists.

e. Type in:

```
$ VAR=abcd
```

```
$ echo Variable is $VAR
```

What is the result after return? Why?

Variable is abcd

because the \$ symbol is the system signal to replace the variable with its value

```
$ echo Variable is $VAR1234
```

What is the result after return? Why?

Variable is

because there is no variable named \$VAR1234

```
$ echo Variable is ${VAR}1234
```

What is the result after return? Why?

Variable is abcd1234

because the {} brackets interpolates the variable, and because of that interpolation it knows to add the other characters afterward.

f. Type in:

```
$ { date; top -b -n1 | head ; } >logfile
```

```
$ cat logfile
```

What is the result after the return? Why?

Wed Apr 12 08:11:08 PM PDT 2023

top - 20:11:08 up 4:39, 1 user, load average: 0.62, 0.51, 0.52

--	--	--

Oregon Institute of Technology

CST 240 Linux Programming Lab #1: Bash, C Review, Primes

Tasks: 314 total, 1 running, 313 sleeping, 0 stopped, 0 zombie
%Cpu(s): 1.4 us, 1.4 sy, 0.0 ni, 97.1 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 15624.9 total, 253.0 free, 3585.9 used, 11786.0 buff/cache
MiB Swap: 2048.0 total, 2047.2 free, 0.8 used. 10687.9 avail Mem

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
5763	cari	20	0	10.9g	425952	116564	S	6.2	2.7	3:38.56	Isolate+
20779	cari	20	0	13208	4116	3256	R	6.2	0.0	0:00.02	top
1	root	20	0	166276	11436	8016	S	0.0	0.1	0:01.97	systemd

because the interpolation command asked for the date, the top command (which displays linux processes) -b :Batch-mode operation

Starts top in Batch mode, which could be useful for sending output from top to other programs or to a file. In this mode, top will not accept input and runs until the iterations limit you've set with the '-n' command-line option or until killed.

and then the -n1 is the number of iterations (one), and then it outputs to the file "logfile" created by that command, and outputs its contents to the terminal because of the cat command.

4. (15 pts) In directory HW1, make a file named "HW1_sample_script.sh". Make sure to copy exactly content with correct spaces. Content of the file is:

```
#!/bin/bash
cat > sample.txt << EOF
cat
dog
spider
anteater
EOF
```

```
cat sample.txt | sort > sorted_sample.txt
cat sorted_sample.txt
```

Afterward, type the command:

```
$ chmod 777 HW1_sample_script.sh
```

Then:

```
$ ./HW1_sample_script.sh
```

--	--	--

Oregon Institute of Technology

CST 240 Linux Programming Lab #1: Bash, C Review, Primes

Show the output you see after executing

```
$ ./HW1_sample_script.sh
```

```
anteater  
cat  
dog  
spider
```

The questions below require some basic research (20 Pts)

5. What is at least one way hackers can break into Unix security.

If a program creates a temporary file in your `/tmp/` directory and it is possible to predict the name of the file it is going to create, then it may be possible to create that file in advance or quickly modify it without the program's knowledge. Programs that create temporary files in a predictable fashion or those that do not set correct permissions (with exclusive access) to temporary files are liable to be exploited. For instance, if a program running as superuser truncates a file `/tmp/9260517.TMP` and it was possible to predict that file name in advance, then a hacker could create a symlink to `/etc/passwd` of the same name, resulting in the superuser program actually truncating the `passwd` file.

6. Explain how you would make a "best" password.

make it unique to all of your other passwords, encrypt it, replace letters with numbers/characters, use a combination of uppercase & lowercase

7. Explain briefly and draw a block diagram of a Unix shell. DO NOT use the same diagram shown in class.

hardware → operating system → shell → user

the operating system lays on the hardware, and the user uses the shell to communicate with the operating system.

Part 2: C Review, primes

Be sure to review the CST 116/126/136/120 concepts

To make sure everyone is on a level playing field, this lab uses C programming on a Linux OS to review. I say again, you MUST use Unix/Linux with vi/emacs/vim... to write source code and gcc to compile and gdb to troubleshoot.

--	--	--

	<h2>Oregon Institute of Technology</h2>	
--	---	--

	<h3>CST 240 Linux Programming Lab #1 : Bash, C Review, Primes</h3>	
--	--	--

This lab will depart from the normal lecture presentations used in previous CST classes. Instead, the students are steered to web sites for reviews of materials learned in freshman year. The ability to research and gain knowledge from web sites is paramount in a software engineer's future.

ALL LABS in CST 240 will be posted on Canvas. Place all source code (copy and paste) and a copy of the test runs (screen shot) into this file.

PROCEDURES

1. Place all your Lab 1 work in your L1 folder.
2. Use vi/emacs/vim... to write a C program.
3. For command line arguments, check out this link:
<https://www.geeksforgeeks.org/command-line-arguments-in-c-cpp/>
4. Watch out. You'll need to convert string arguments to integers (atoi!)
5. For file IO in C, check this link: <https://beginnersbook.com/2014/01/c-file-io/>
6. To compile and execute a C program in Linux, check out this link:
http://akira.ruc.dk/~keld/teaching/CAN_e14/Readings/How%20to%20Compile%20and%20Run%20a%20C%20Program%20on%20Ubuntu%20Linux.pdf You may need to force compilation with math functions with: -lm
7. There are many more links, YouTube videos on running C program in Linux. Check it out yourself if needed.
8. Labs will be demonstrated in lab.

1: Prime Numbers. (50 Pts)

This program uses command line argument to input a string representing an integer. In the command line, one can only input a string so integer 1234 can only be input as "1234". atoi is needed to convert "1234" to 1234.

For command line arguments, check out link:

<https://www.geeksforgeeks.org/command-line-arguments-in-c-cpp/>

The program will output the input number in words format then "IS Prime!" or "is NOT Prime!" depending on whether the input integer is a prime or not. For clarification, check out examples below.

Example output to screen if executable is named prime.out

--	--	--

	<h2>Oregon Institute of Technology</h2>	
--	---	--

	<h3>CST 240 Linux Programming Lab #1: Bash, C Review, Primes</h3>	
--	---	--

`$/prime.out three`

Got to enter string representing integer only ("235", "666"...), try again!

`$/prime.out 120`

One Two Zero is NOT Prime!

`$/prime.out 1031`

One Zero Three One IS Prime!

Grading:

Code working correctly: 25 pts

Code documentation: 10 pts

Code testing: 15 pts

Total: 50 pts

For documentation, be sure to include your name, the date, the class and assignment at the top.

Also be sure to describe what every major section of code is doing in a comment.

When testing, be sure to test boundary conditions and unexpected data along with expected data. You should have at least 5 different tests.

--	--	--