Final Exam

Due Dec 9 at 11:59pm **Points** 39.96 **Questions** 37
Available Dec 6 at 8am - Dec 11 at 11:59pm 6 days **Time Limit** 120 Minutes

Instructions

Finish this test before the **due date**. Once you begin, you will have 120 minutes to complete your work. You must complete the exam is a single session, please plan your time accordingly. If you do not submit before that time your incomplete exam will be automatically submitted as is.

You can look at lectures, lab assignments, texts and even use your Linux account, but you may not consult any other individuals for help.

Each question is worth 1.08 points, for an exam total of about 40 points.

Multiple choice questions with square check-boxes may have more than one correct answer. Multiple choice questions with round radio-buttons have only one correct answer.

Any code fragments you are asked to analyze are assumed to be contained in a program that has all the necessary variables defined and/or assigned.

Attempt History

	Attempt	Time	Score
LATEST	Attempt 1	111 minutes	36.54 out of 39.96

Score for this quiz: **36.54** out of 39.96
Submitted Dec 9 at 11:17pm
This attempt took 111 minutes.

Question 1 0.54 / 1.08 pts

$ cat colors
red
red
green
purple
green

Given the file **colors** as shown above, what filter(s) will purge duplicates displayed to standard output? (Select any that apply.)

- [ ] $ cat colors | uniq

Correct!

- [x] **$ sort colors | uniq**

You Answered

- [x] **$ cat colors | uniq -u**

Correct!

- [x] **$ sort -d colors | uniq**

Hints

Feedback

Key: B, D

$ cat colors
red
red
green
purple
green

duplicates purged
$ sort colors | uniq
green
purple
red
$ sort -d colors | uniq
green
purple
red

non adjacent duplicates retained
$ cat colors | uniq
red
green
purple
green

$ cat colors | uniq -u
green
purple
green

Question 2

1.08 / 1.08 pts

Given the child script as follows:

```
# !/bin/bash
echo "==>Entering child process<=="
apples=green
bananas=black
echo "==Showing fruit variables in child<=="
echo "apples are $apples and bananas are $bananas"
```

Given an alias written as follows:

$ alias fruit='echo apples are \"$apples\" and bananas are \"$bananas\"'

Given the parent variables defined as follows:

$ apples=red; bananas=yellow

Given the sequence:
1: **$ ./child**
2: **$ fruit**

What will be the output after line 2:?

- apples are "green" and bananas are "black"

Correct!

- apples are "red" and bananas are "yellow"
- apples are green and bananas are black
- apples are red and bananas are yellow

Feedback

$ cat child
!/bin/bash
echo "==>Entering child process<=="
apples=green
bananas=black
echo "==Showing fruit variables in child<=="
echo "apples are $apples and bananas are $bananas"

$ child
==>Entering child process<==
==Showing fruit variables in child<==
apples are green and bananas are black

$ fruit
apples are "red" and bananas are "yellow"

The child cannot change the parent's variables.

Question 3

1.08 / 1.08 pts

Given the following listing for the file **permslice** what will be the output of this sequence after line 3:?

1: **$ ls -l permslice**
-rwx------ 1 glamble dip 0 Nov 16 18:17 permslice

2: **$ vcut=$(ls -l permslice | cut -d " " -f1)**

3: **$ echo $vcut**

- 1
- -r
- -rwx

Correct!

-rwx------

Feedback

This command cuts 1st field (with delimiter " " separating fields).

$ ls -l permslice
-rwx------ 1 glamble dip 0 Nov 16 18:08 permslice
$ vcut=$(ls -l permslice | cut -d " " -f1)
$ echo $vcut
-rwx------

Question 4

1.08 / 1.08 pts

How can you keep other online users from using the **write** command to communicate with you when you are logged on?

Correct!

$ mesg n

$ news -n

$ mesg y

None of the above

Feedback

Set the mesg command to NO to stop receiving messages.

$ mesg n

Question 5

1.08 / 1.08 pts

In a regular expression a **.** (dot) matches a *single* character.

Correct!

- (•) True
- ( ) False

Feedback

The character "**.**" (dot) is a special regular expression *meta-character*. By itself it will match any **single** character, except the end-of-line character.

Question 6

1.08 / 1.08 pts

The command to list *all* the files that have filenames that start with the letter p and end with a number from 1 to 3 is _____.

- ( ) $ ls p?1-3
- ( ) $ ls p[1-3]*

Correct!

- (•) **$ ls p*[1-3]**
- ( ) $ ls p[3]

Feedback

$ ls p*[1-3]

This command anchors p as the starting letter allowing any (including none) subsequent letters ending with a number 1, 2 or 3.

Example:

$ ls p*[1-3]
p1 p2 pabc2 paul2 pc3 pop3

Question 7

1.08 / 1.08 pts

Given the file **mathex** as follows:

$ cat mathex
2 + 2
12 * 3
5/0

What will be the result of redirecting **stdin**, **stdout** and **stderr** as shown in the command line below?

$ bc < mathex > answers 2>errors

answers file will contain

4

36

errors file will be empty

answers file will be empty

errors file will contain an error message

Correct!

answers file will contain

4

36

errors file will contain an error message

The **answers** file will not be created

The **errors** file will contain

4

36

and an error message

Feedback

bc command reads input from redirected **mathex** file, writes standard output to the **answers** file and stdard error to the **errors** file.

$ cat answers
4
36
$ cat errors
Runtime error (func=(main), adr=5): Divide by zero

Question 8

1.08 / 1.08 pts

$ grep -r tomato /usr 2>&1 /dev/null &

$ ps

```
 PID  TTY      TIME     CMD
1567 pts/0    00:00:00 bash
2265 pts/0    00:00:01 grep
2266 pts/0    00:00:00 ps
```

Given the snapshot of the currently running processes shown above how can you terminate the background **grep** command?

- $ fg 2265
- $ sleep 2265
- $ kill 2266
- **$ kill 2265**

Correct!

Feedback

To stop the background process running the **grep** command use the asynchronous signal **kill** <job number>.

$ ps

```
  PID  TTY      TIME     CMD
 1567  pts/0    00:00:00 bash
 2265  pts/0    00:00:01 grep
 2266  pts/0    00:00:00 ps
```

$ kill 2265

```
[1]+  Terminated              grep -r potato /usr /dev/null 2>&1
[glamble@linux60812 final]$ ps
 PID  TTY      TIME     CMD
1567 pts/0    00:00:00 bash
2280 pts/0    00:00:00 ps
```

Question 9

1.08 / 1.08 pts

Given the following contents of the **todo** file as shown below:

$ cat todo
groceries
bills
laundry
exercise
haircut
bank

Use **sort** as a filter, to rewrite the following sequence of commands:

$ sort todo > someday

$ head -n2 someday

$ rm someday

Correct!

- (•) **$ cat todo | sort | head -n2**
- ( ) **$ sort today | someday | head -n2**
- ( ) **$ cat todo | someday | sort**
- ( ) **$ cat todo | sort | head**

Hints

Feedback

$ sort todo > someday
$ head -n2 someday
bank
bills
$ rm someday

The above sequence of commands can be rewritten as a filter using **sort:**

$ cat todo | sort | head -n2
bank
bills

Question 10

1.08 / 1.08 pts

Consider the following error message this user received trying to run **bdayscript.** What might be a logical next step to troubleshoot why the script will not run?

$ ls -l bdayscript
-rwxr-x--- 1 glamble dip 0 Nov 15 11:29 bdayscript
$ bdayscript
-bash: bdayscript: command not found

Enable execute permission for group and others.

$ chmod -v 777 bdayscript
mode of `bdayscript' changed to 0777 (rwxrwxrwx)

Correct!

Check the environment variable setting to see if the current directory is on the user's path.

$ pwd
/home/glamble/cs30a/bin
$ echo $PATH

Feedback

Oops, the current directory is not on the user's path:
$ pwd
/home/glamble/cs30a/bin

$ echo $PATH
/sbin:/bin:/usr/sbin:/usr/bin:/usr/local/sbin:/usr/local/bin:/usr/lib:/usr/bin/X11:/usr/local/bin:/opt/gnu/bin

Question 11

0.54 / 1.08 pts

Using the **find** utility, peform the following task:

List the inode numbers of all files in the current working directory whose filenames end in **.c**. (List all that apply.)

- ☐ **$ find . -inum "*.c"**
- ☐ **$ find . "*.c" -inum**
- Correct Answer ☐ **$ find . -name "*.c" -ls**
- Correct! ☑ **$ find . -name '*.c' -ls**

Feedback

Key: C, D

Examples:

```
$ find . -name "*.c" -ls
2826470324    0 -rw-------   1 glamble  dip          0 Nov 19 15:19 ./p.c
2826470325    0 -rw-------   1 glamble  dip          0 Nov 19 15:19 ./a.c
$ find . -name '*.c' -ls
2826470324    0 -rw-------   1 glamble  dip          0 Nov 19 15:19 ./p.c
2826470325    0 -rw-------   1 glamble  dip          0 Nov 19 15:19 ./a.c
```

Question 12

1.08 / 1.08 pts

The following shell variables have been created as follows:

$ penny=one; nickel=five; dime=ten

How can you print the values of the shell variables at the command line? (Select all that apply.)

- [ ] $ echo penny nickel dime
- [ ] $ echo '$penny' '$nickel' '$dime'
- [x] **$ echo "$penny" "$nickel" "$dime"** — Correct!
- [x] **$ echo $penny $nickel $dime** — Correct!

Feedback

prints the names of the shell variables
$ echo penny nickel dime
penny nickel dime

$ echo '$penny' '$nickel' '$dime'
$penny $nickel $dime

print the values of the shell variables
$ echo "$penny" "$nickel" "$dime"
one five ten

print the values of the shell variables
$ echo $penny $nickel $dime
one five ten

Question 13

0.54 / 1.08 pts

Say the user tries to run the script **moonwalk** receiving the error message below:

$ ./moonwalk
-bash: ./moonwalk: Permission denied
$ ls -l moonwalk
-rw------- 1 glamble dip 0 Nov 19 10:01 moonwalk

How can the user enable execute permissions for the script file **moonwalk**? (Select all that apply.)

You Answered

- [x] **$ add +x moonwalk**
- [ ] **$ chmod 600 moonwalk**

Correct!

- [x] **$ chmod 700 moonwalk**

Correct!

- [x] **$ chmod u+x moonwalk**

Feedback

Key: C, D

Using the **chmod** command, the user can add the mnemonic setting for execute permission or set the execute bit numerically.

$ ls -l moonwalk
-rw------- 1 glamble dip 0 Nov 19 10:01 moonwalk
$ chmod 700 moonwalk
$ ls -l moonwalk
-rwx------ 1 glamble dip 0 Nov 19 10:01 moonwalk

$ ls -l moonwalk
-rw------- 1 glamble dip 0 Nov 19 10:01 moonwalk
$ chmod +x moonwalk
$ ls -l moonwalk
-rwx------ 1 glamble dip 20 Nov 19 10:04 moonwalk

$ ./moonwalk
to the moon!

Question 14

1.08 / 1.08 pts

$ ps -l

F	S	UID	PID	PPID	C	PRI	NI	ADDR	SZ	WCHAN	TTY	TIME	CMD
0	S	1452	1567	1566	0	80	0	-	28856	wait	pts/0	00:00:00	bash
0	T	1452	3514	1567	0	80	0	-	25227	signal	pts/0	00:00:00	sleep
0	R	1452	3515	1567	0	80	0	-	27032	-	pts/0	00:00:00	ps

What is the parent process ID of the running command?

- [ ] 3515
- [ ] 3514

1566

Correct!

1567

Feedback

$ ps -l

F	**S**	UID	PID	**PPID**	C	PRI	NI	ADDR	SZ	WCHAN	TTY	TIME	**CMD**
0	S	1452	1567	1566	0	80	0	-	28856	wait	pts/0	00:00:00	bash
0	T	1452	3514	1567	0	80	0	-	25227	signal	pts/0	00:00:00	sleep
0	**R**	1452	3515	**1567**	0	80	0	-	27032	-	pts/0	00:00:00	**ps**

The parent process ID (PPID) of the running (S=R) **ps** command is 1567.

Question 15

1.08 / 1.08 pts

$ echo $PATH

/sbin:/bin:/usr/sbin:/usr/bin:/usr/local/sbin:/usr/local/bin:/usr/lib:/usr/bin/X11:/usr/local/bin:/o

What is the fourth directory on this path?

/usr/sbin

Correct!

/usr/bin

/usr/local/sbin

/usr/local/bin

Feedback

The **":"** delimiter separates the directories.

Question 16

0.72 / 1.08 pts

Using the **grep** command with the **-n** option how can you find the line number for the quote "Like a diamond in the sky" in the file **twinkle**? (Select all that apply.)

$ cat twinkle
Twinkle, twinkle, little star,
How I wonder what you are!
Up above the world so high,
Like a diamond in the sky.

Correct! **$ grep -n 'Like a diamond in the sky' twinkle**

Correct! **$ grep -n "Like a diamond in the sky" twinkle**

You Answered **$ grep -n Like a diamond in the sky < twinkle**

Correct! **$ grep -n "Like a diamond in the sky" < twinkle**

Feedback

Key: A, B, D

$ grep -n 'Like a diamond in the sky' twinkle
4:Like a diamond in the sky.

$ grep -n "Like a diamond in the sky" twinkle
4:Like a diamond in the sky.

$ grep -n "Like a diamond in the sky" < twinkle
4:Like a diamond in the sky.

Question 17

1.08 / 1.08 pts

You can execute the shell script **sharktank** with the following permission settings on the file.

Example:

$ ls -l sharktank
--wx------ 1 glamble dip 17 Nov 16 17:01 sharktank

- True

Correct!

- False

Feedback

Because the shell must read the commands from the file containing a shell script before it can execute the commands, you must have read permission for the file to execute a shell script.

Question 18

1.08 / 1.08 pts

$ cat sedq
Maybe I can do it.
Maybe you can do it.
Maybe it will be sunny.
Maybe I will get a dog for Christmas.

Using **sed** how can you substitute each Maybe as seen in the lines of the file sedq to Definitely displaying the result to standard output?

- **$ sed s /Maybe/Definitely/ sedq**
- **$ sed '1s/Maybe/Definitely/' sedq**

Correct!

- **$ sed 's/Maybe/Definitely/' sedq**

$ sed '2,4s/Maybe/Definitely/' sedq

Hints

Feedback

Use **the sed substitute command, s,** to substitute pattern Maybe for replacement Definitely.

Question 19

1.08 / 1.08 pts

Given the parent variables defined as follows:

$ apples=red; bananas=yellow

Consider the script file **child** as shown below:

$ cat child

```
# !/bin/bash
echo "==>Entering child process<=="
echo "==Showing fruit variables in child<=="
echo "apples are $apples and bananas are $bananas"
```

Will the child script see the values of the apples and bananas variables? (Select True for yes; Select False for no.)

True

Correct!

False

Feedback

$ apples=red; bananas=yellow

$ child
==>Entering child process<==
==Showing fruit variables in child<==
apples are and bananas are

No, the child cannot view the values of the parent's non-exported variables.

Question 20

1.08 / 1.08 pts

How could you issue an alias in the **bash** shell to create a '*nickname*' **h** for the **history** command?

- $ alias h = history
- $ set h=history
- Correct! **$ alias h=history**
- $ unalias h=history

Feedback

Use the **alias** command to activate an alias for an established LInux command.

Question 21

1.08 / 1.08 pts

1: **$ fishbowl**
-bash: fishbowl: command not found
2: **$ ./fishbowl**

How can a path be made so that the **fishbowl** script can be executed directly as in line 1: versus having to use the workaround of using a **./** in front of the command as in 2:?

Add the "." directory to the path with the following command:

- **$ PATH=PATH:.**
- Enable execute permission on the fishbowl script file.
- Enable execute permission on the parent directory of the fishbowl file.

Correct!

Add the **"."** directory to the path with the following command:

- **$ PATH=$PATH:.**

Feedback

Add the current directory (**"."**) to the path with the following command:

$ PATH=$PATH:.
$ fishbowl
clean the fishbowl, please!

Question 22

1.08 / 1.08 pts

How could you determine the number of commands in /bin that are really scripts?

- **$ file /bin**

Correct!

- **$ file /bin/* | grep script | wc -l**
- **$ type /bin/* | grep script | wc -l**

None of the above

Feedback

This filter will give you the number of commands in /bin that are really scripts.

$ file /bin/* | grep script | wc -l
8

Question 23

1.08 / 1.08 pts

Given the following contents of the script file **tvset** what will be the output of the command at line 3?

$ cat tvset
#!/bin/bash
echo "Turn the tv on or off? "
read tv
echo "tv is set to $tv"
exit

1: **$ tv=off; export tv**
2: **$ ./tvset**
Turn the tv on or off?
on
tv is set to on
3: **$ echo $tv**

$tv

on

Correct!

off

tv

Feedback

The child process is unable to change the value of the parent variable.

$ cat tvset
#!/bin/bash
echo "Turn the tv on or off? "
read tv
echo "tv is set to $tv"
exit

1: **$ tv=off; export tv**

2: **$ ./tvset**
Turn the tv on or off?
on
tv is set to on

3: **$ echo $tv**
off

Question 24

0 / 1.08 pts

Given the system process status display below what is the PID and the status of the **bash** command?

$ ps -l -u glamble

```
F S   UID   PID    PPID  C PRI  NI ADDR SZ WCHAN  TTY      TIME     CMD
5 S  1452 17536 17527  0  80   0 - 25018 poll_s  ?        00:00:00 sshd
0 S  1452 17537 17536  0  80   0 - 28868 wait    pts/1    00:00:00 bash
0 S  1452 20343 17537  0  80   0 - 25227 hrtime  pts/1    00:00:00 sleep
0 R  1452 20344 17537  0  80   0 - 29141 -       pts/1    00:00:00 ps
```

You Answered

- PID = 17537; status is running

Correct Answer

- PID = 17537; status is sleeping
- PID = 17536; status is stopped
- PID = 17536; status is sleeping

Feedback

The bash command PID = 17537; command status is sleeping (S).

F **S** UID **PID** PPID C PRI NI ADDR SZ WCHAN TTY TIME CMD
0 **S** 1452 **17537** 17536 0 80 0 -28868 wait pts/1 00:00:00 bash

Question 25

1.08 / 1.08 pts

What are the numerical permissions for the file **cando**?

$ **ls -l cando**
-r-xrw-rwx 1 glamble dip 0 Nov 19 09:43 cando

- 273

Correct!

- 567
- 356
- 133

Feedback

$ ls -l cando
-r-xrw-rwx 1 glamble dip 0 Nov 19 09:43 cando

You can decode the numerical permission bits for the file as follows:

d|r-x|rw-|rwx => 101 | 110 | 111 => 567

Question 26

1.08 / 1.08 pts

Assume you have made the following assignment:

$ person=jake

Match the corresponding output to each of the following commands.

Correct! **$ echo $person** — jake

Correct! **$ echo '$person'** — $person

Correct! **$ echo 'hello $person'** — hello $person

Correct! **$ echo "hello $person"** — hello jake

Correct! **$ echo "hello '$person'"** — hello 'jake'

Feedback

$ person=jake

$ echo $person
jake

$ echo '$person'
$person

$ echo 'hello $person'
hello $person

$ echo "hello $person"
hello jake

$ echo "hello '$person'"
hello 'jake'

Question 27

1.08 / 1.08 pts

Given the directory contents:

```
$ ls -l
total 4
-rw------- 1 glamble dip   0 Feb 18 05:00 testfile
-r-x------ 1 glamble dip 108 Feb 18 05:03 testwrite

$ cat testwrite
#!/bin/bash
if test -w "$1"
  then
  echo "file $1 is write-able"
  else
  echo "cannot write to $1"
fi
```

Considering the testwrite script shown above what will be the result of the test command shown below?

```
$ ./testwrite nofile
```

Correct!

- (•) cannot write to nofile
- ( ) file $1 is write-able
- ( ) file testfile is write-able
- ( ) error - nofile does not exist

Feedback

Given the directory contents:

```
$ ls -l
total 4
-rw------- 1 glamble dip   0 Feb 18 05:00 testfile
-r-x------ 1 glamble dip 108 Feb 18 05:03 testwrite

$ cat testwrite
#!/bin/bash
if test -w "$1"
   then
   echo "file $1 is write-able"
   else
   echo "cannot write to $1"
fi

$ ./testwrite nofile
cannot write to nofile
```

Question 28

1.08 / 1.08 pts

Is each of the following an absolute pathname, a relative pathname, or a simple filename?

Correct! **gotmilk** — simple filename

Correct! **customer/bills/gotmilk** — relative pathname

Correct! **/usr/lib** — absolute pathname

Feedback

The absolute pathname is the location of the argument relative to the **root** directory. All absolute pathnames begin with a slash (*/*). A relative pathname is the directory path relative to the **current working** directory. Relative pathnames do not begin with a slash (*/*).

Question 29

1.08 / 1.08 pts

Match the file redirection descriptions below.

Correct!	**to redirect stdin**	either 0< or just <
Correct!	**to redirect stdout**	either 1> or just >
Correct!	**to redirect stderr**	2> filename
Correct!	**to redirect stdout and append**	>> filename

Feedback

The input and output of a program can be redirected from and to other files using **<**, **>**, **2>** and **>>**:

0< filename or **<** filename # To redirect **stdin** (either **0<** or just **<**)

1> filename # To redirect **stdou**t (either **1>** or just **>**)

2> filename # To redirect **stderr**

>> filename # To redirect **stdout** and append

Question 30

1.08 / 1.08 pts

The **kernel** manages all the hardware dependent functions in the Linux operating system.

Correct!

- (•) True
- ( ) False

Question 31

1.08 / 1.08 pts

How can you cause **vi** to enter Input mode? (Select all that apply.)

Correct!

- [x] **A**

Correct!

- [x] **a**

Correct!

- [x] **I**

Correct!

- [x] **i**

Correct!

- [x] **O**

Correct!

- [x] **o**
- [ ] **w**
- [ ] **W**

Feedback

Key: A, B, C, D, E, F

Any of the following commands will put vi in Input mode: **A, a, I, i, O, and o**.

Example:

~
~
-- INSERT --

Question 32

1.08 / 1.08 pts

$ cat mysteryfile
haunted house
goblins
ghosts
bats
spiders

Given the original **mysteryfile** as shown above, what is the content of the **mysteryfile** after the following command is executed?

$ date > mysteryfile

- haunted house
  goblins
  ghosts
  bats
  spiders
  date

- date
- Correct! output of date command
- empty file

Feedback

The stdout redirection (**>**) results in the output of the **date** command overwriting the original file contents of the mysteryfile.

Question 33

1.08 / 1.08 pts

Linux looks at the devices such as printers and terminals as files.

- Correct! True
- False

Feedback

Question 34

1.08 / 1.08 pts

List the commands you can use to perform these operations in the order shown below: (Select all that apply.)

1: Make your home directory the current directory

2: Identify the working directory

- [ ] `$ cd /; pwd`

Correct!

- [x] `$ cd; pwd`

Correct!

- [x] `$ cd ~; pwd`

Correct!

- [x] `$ cd $HOME; pwd`

Feedback

Key: B, C, D

Note: A takes you to the root directory.

Question 35

1.08 / 1.08 pts

The **mv** and **cp** commands are the same. Both copy file(s) from one place to another leaving both the source and destination arguments intact.

Example: **cp src dst** versus **mv src dst**

- True

Correct!

- False

Feedback

The **mv** command is destructive versus the **cp** command leaves the **src** argument intact.

Example:

$ mv src dst # src no longer exists

$ cp src dst # src intact

Question 36

0.72 / 1.08 pts

What is a Linux shell? (Select all that apply.)

- firewall

Correct!

- command interpreter

Correct Answer

- programming language (i.e. can run files of commands)

Correct!

- a process

Feedback

Key: B, C, D

The Linux shell is a process that acts as a command interpreter that can also be used as a programming language to run files of commands (shell scripts).

Question 37

1.08 / 1.08 pts

The command to create a directory called **cs30b** is _____.

Correct!

- (•) **$ mkdir cs30b**
- ( ) $ md cs30b
- ( ) $ mk cs30b dir
- ( ) $ dir cs30b

Feedback

Next quarter: hope you'll have the need to create a directory called cs30b!

Quiz Score: **36.54** out of 39.96