**CSc 3320: Systems Programming**

Fall 2021

Midterm 1: Total points = 100

Submission instructions:

1. Create a Google doc for your submission.

2. Start your responses from page 2 of the document and copy these instructions on page 1.

3. Fill in your name, campus ID and panther # in the fields provided. If this information is missing TWO POINTS WILL BE DEDUCTED.

4. Keep this page 1 intact. If this *submissions instructions* page is missing in your submission TWO POINTS WILL BE DEDUCTED.

5. Start your responses to each QUESTION on a new page.

6. If you are being asked to write code copy the code into a separate txt file and submit that as well. The code should be executable. E.g. if asked for a C program then provide myfile.c so that we can execute that script. In your answer to the specific question, provide the steps on how to execute your file (like a ReadMe).

7. If you are being asked to test code or run specific commands or scripts, provide the evidence of your outputs through a screenshot and/or screen

video-recordings and copy the same into the document.

8. Upon completion, download a .PDF version of the google doc document and submit the same along with all the supplementary files (videos, pictures, scripts etc).

9. Scripts/Code without proper comments, indentation and titles (must have the name of the program, and name & email of the programmer on top the script).

Full Name: Adrian Rivas

Campus ID: arivas7

Panther #: 002-39-5009

**Questions 1-5 are 20pts each**

1. (20 pts) Pick any of your 10 favourite unix commands. For each command run the *man* command and copy the text that is printed into a mandatabase.txt. Write a shell script *helpme.sh* that will ask the user to type in a command and then print the manual’s text associated with that corresponding command. If the command the user types is not in the database then the script must print

*sorry, I cannot help you*

Text

Description automatically generated

Text

Description automatically generated

A screenshot of a computer

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

#!/bin/bash

echo "Type a command"

read cmd\_in

if grep -q $cmd\_in mandatabase.txt; then

grep $cmd\_in mandatabase.txt;

else

echo "sorry, I cannot help you"

fi

2. (10pts each) On your computer open your favourite Wikipedia page. Copy the text from that page into a text file **myexamfile.txt** and then copy that file to a directory named **midterm** (use mkdir to create the directory if it doesn’t exist) in your snowball server home directory (use any FTP tool such as Putty or Filezilla to copy the file from your computer to the remote snowball server machine: see Lab 6).

a. Write a shell script that will find the number of statements in the text. A statement is defined as the collection of text between two periods (full-stops).

Text

Description automatically generated

Graphical user interface

Description automatically generated with medium confidence



b. Update the script to present a tabular list that shows the number of words and number of letters in each statement.

Text

Description automatically generated

A picture containing text

Description automatically generated

#!/bin/bash

function find\_statements(){ #fucntions to find the word in the file

i=0

while IFS=. read -r -a line #will call the array and the seperations

do

IFS="."

for stat in ${line[@]} # the line of the array

do

statements[i]=$(echo $stat | sed 's/^ //g') #assigns the parts of the array

i=$((i+1)) # this will increase the number in the word count

done

done < $1

echo -e "\t\tWords\tletters" # will show the header of the word count

i=1 # start the array fro the statment to counter the first array

for statement in ${statements[@]} # Will show statements in the array

do

words=$(echo $statement | wc -w) # word count

letters=$(echo $statement | tr -d ' ' | wc -c) # letter count

echo -e "Statement $i:\t $words\t $((letters-1))" #show the word and letter count

i=$((i+1))

done

}

if(($#==1)) # will check in the user input is right

then

file=$1

else

echo -n "Text file: "

read file

fi

if [ -f $file ]

then

find\_statements $file # calls the file in the statement

else # would say error if it does not exist

echo "$file: file not exist"

exit 1

fi

3. (20pts) Design a calculator using a shell script using regular expressions. The calculator, at the minimum, must be able to process addition, subtraction, multiplication, division and modulo operations. It must also have cancel and clear features.

Text

Description automatically generated

#!/bin/bash

echo "Enter Numbers: "

read first

echo "Enter Numbers:"

read second

echo "Want to Exit: "

read cancel

echo "Want to erase: "

read erase

#input the type of operations for the calculator

echo "Choose one:"

echo "1. Addition"

echo "2. Subtraction"

echo "3. Multiplication"

echo "4. Division"

echo "5. Cancel"

echo "6. Clear"

read ch

# read the operation and the calculator expressions

case $ch in

1) res=`echo $first + $second | bc`;;

2) res=`echo $first - $second | bc`;;

3) res=`echo $first \\* $second | bc`;;

4) res=`echo "scale=2; $first /$second" | bc` ;;

5) res=`echo $cancel "exit=1" | bc` ;;

6) res=`echo $erase | bc`;;

esac

echo "Result: $res"

exit 0

Text

Description automatically generated

Text

Description automatically generated with medium confidence

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

4. (20pts) Build a phone-book utility that allows you to access and modify an alphabetical list of names, addresses and telephone

numbers. Use utilities such as awk and sed, to maintain and edit the file of phone-book information. The user (in this case, you) must be able to read, edit, and delete the phone book contents. The permissions for the phone book database must be such that it is inaccessible to anybody other than you (the user).

Text

Description automatically generated

#!/bin/bash

BOOK = "my\_phonebook.txt"

touch $BOOK #create a blank file

chmod 700 $BOOK #permission for the user

exit=0 #exit the flag

while [ $exit -ne 1 ] #run the flag until pressed 1

do

echo -e "Enter 1 to add, 2 to edit, 3 to view, 4 delete, 5 to exit : "

read answer #instruction of the file

if [ "$answer" = "1" ] # add new entry in the file

then

echo "Enter name: "

read name

echo "Enter Adrress: "

read addr

echo "Enter phone number: "

read phone

cat $BOOK > temp #copies the previous entries

echo $name $addr $phone >> temp #focus on the new entries

sort -nk1 temp > $BOOK #sort the name in alphatetical order

elif [ "$answer" = "2" ] #this would edit the old entry

then

echo "Enter name to edit: "

read name

cp $BOOK temp

grep -vw $name $BOOK > temp #pick all the entries except the one that was edit

echo "Enter new name: "

read name

echo "Enter new address: "

read addr

echo "Enter new phone number: "

read phone

echo $name $addr +$phone >> temp #focues on the entry that is goign to be edit

sort -nk1 temp > $BOOK # sort and overwrite the phonebook

elif [ "$answer" = "3" ] # display the entry that would exit

then

cat $BOOK # show the content

elif [ "$answer" = "4" ] # deletes the entry

then

echo "Enter name to delete: "

read name

cp $BOOK temp

grep -vw $name $BOOK > temp #picks the entries but the one that gets deleted from the file

sort -nk1 temp > $BOOK # sorts whats in temp

elif [ "$answer" = "5" ] #exit option in the file

then

exit=1 # sets exit the file

else

echo "command not available!" #message invalid command

fi

done

exit 0

Text

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

5. (4 pts each) Give brief answers with examples, wherever relevant

A. What is the use of a shell?

A shell is the operating system’s command line interface and that the system that interprets for the set of commands that the user who is you set in the system and are used to translate and to communicate to the terminal.

B. Is there any difference between the shell that you see on your PC versus that you see on the snowball server upon login. If yes, what are they? Provide screenshots for examples.

I think they are a difference because in the snowball server is a shell that connected to a server to store information and keep it together with users so they can share and only users with permission can login into the shell. In the PC, does not store the information together where the user can later get the file, and anyone can access it.

Text

Description automatically generated

C. What are the elements in a computer (software and hardware) that enable the understanding and interpretation of a C program?

In the computer the program C is a compiled language that a compiler interprets C and is generates it into machine code which the computer can read the language and then vice versa to switch back so you can read the language.

D. The “printf()” C command is used for printing anything on the screen. In bash we use the command “echo ”. What is the difference (if any) in terms of how the computer interprets and executes these commands?

The difference between echo and printf is that echo always exits with a zero status and just prints standard outputs and while in printf lets the user for formatting a string and gives a non zero exit status code until it fails.

E. What do these shell commands do? “ssh”, “scp” and “wget”. Describe briefly using an example that you have executed using the snowball server.

SSH: The meaning of sh is that is a secure shell, in this shell it is a secure connection between systems to share, move, or copy files between those systems.

SCP: It is a tool that is used in the SSH network to copy files remotely to the server computer.

WGET: This is a network downloader that is non-interactive to download files by using a URL from the server.