**CSc 3320: Systems Programming**

Spring 2021

Midterm 1: Total points = 100

Assigned: 26th Feb 2021: 12.01 PM

**Submission Deadline: 2nd Mar 2021: 12.01 PM**

**(No extensions. If your submission is not received by this time then it will NOT be accepted.)**

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 script 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).

Full Name: Adam Nguyen

Campus ID: anguyen117

Panther #: 900911012

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

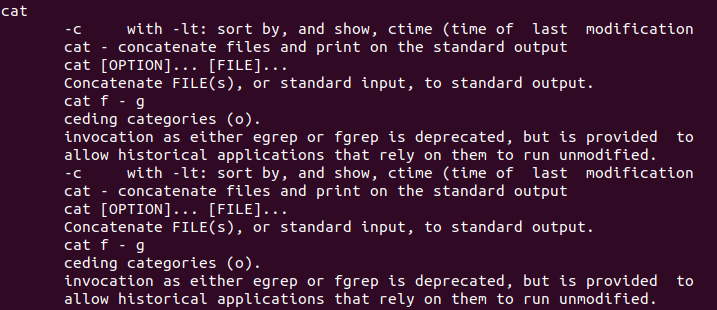
1. Pick any of your 10 favourite unix commands. For each command run the *man* command and copy the text that is printed into a mandatabase.txt. Write a shell script *helpme.sh* that will ask the user to type in a command and then print the manual’s text associated with that corresponding command. If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

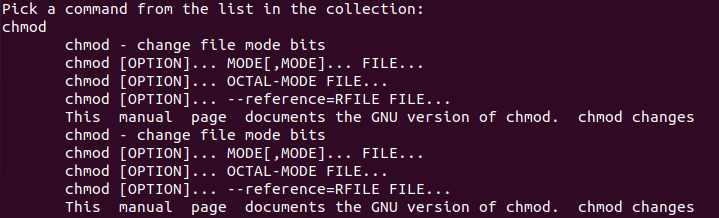
*sorry, I cannot help you*

1)

chmod a+x helpme.sh to give permission to execute helpme.sh

./helpme.sh to execute







Code for helpme.sh: (Can also check the file uploaded)

#!/bin/bash

echo "Pick a command from the list in the collection: "

read input

if (grep $input mandatabase.txt); then

{

grep $input mandatabase.txt;

}

else

echo "sorry, I cannot help you"

fi

2. 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).

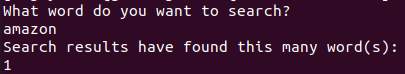
Write a shell script that will find the number of occurrences of a particular keyword typed by the user. Present evidence of your testing with at least 5 trials (different keywords each time)

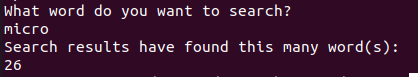
2)

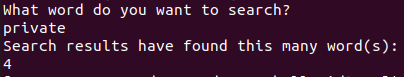
Created a .txt file and placed information from Wikipedia to myexamfile.txt file and saved it to desktop.

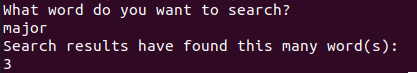
Copied a file using Filezilla from a myexamfile.txt from windows to midterm directory by logging into snowball and selecting the .txt on the left side of the program and selecting the midterm folder on right side and entering it. Transfer the file by double clicking the file.

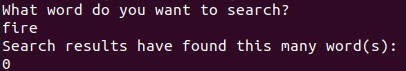
The .sh file is called “fileSearcher.sh” and we have the .txt file called myexamfile.txt.

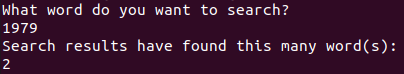












chmod a+x fileSearcher.sh to give permission to execute fileSearcher.sh

./fileSearcher.sh to execute

Code for fileSearcher.sh (can also be found by the uploaded file):

#!/bin/bash

echo "What word do you want to search?"

read word

echo "Search results have found this many word(s): "

grep -o -i $word ~/midterm/myexamfile.txt | wc -l

Myexamfile.txt is too large to be posted here. Refer to the uploaded file to search word counts in it.

3. Write a shell script to find files in a directory hierarchy (e.g. your home directory) that have not been accessed for N days and compress them. Here N is a parameter and the user will be asked for that input as the first step of the script execution.

3)

chmod a+x Compressor.sh to give permission to execute Compressor.sh

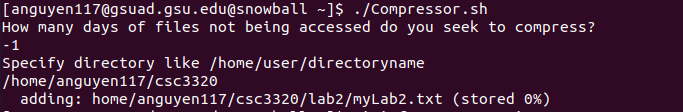
./Compressor.sh to execute

Enter value of N to determine how many days since accessed to compress.

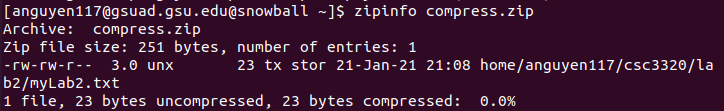
Give the absolute pathway to ensure the input of the directory hierarchy.

Since it never mentioned how to compress it, I will compress it into a .zip file.

It seems that using the script accesses the files, so the parameter N = -1 may be needed to make it so that the results of compressing can be seen.



zipinfo compress.zip to view contents.



Code for Compressor.sh (also in upload file):

#!/bin/bash

echo "How many days of files not being accessed do you seek to compress?"

read N

echo "Specify directory like /home/user/directoryname"

read directory

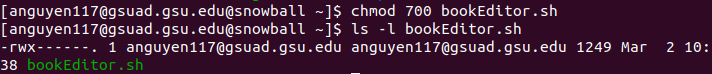
find $directory -iname "\*" -atime +$N -type f | zip compress -@

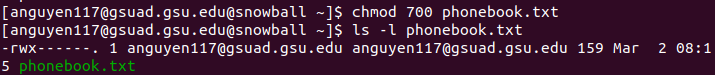
#Since it never mentioned how to compress it, I will compress it into a .zip file.

4. 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 the user.

4)



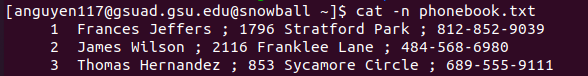


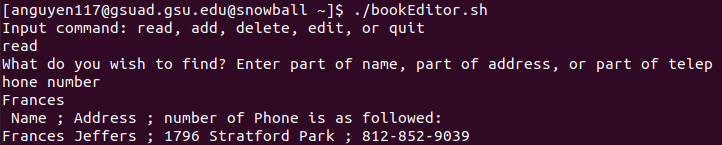
Phone book database is only accessible to the user.

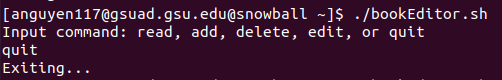
chmod a+x bookEditor.sh to give permission to execute the bookEditor.sh

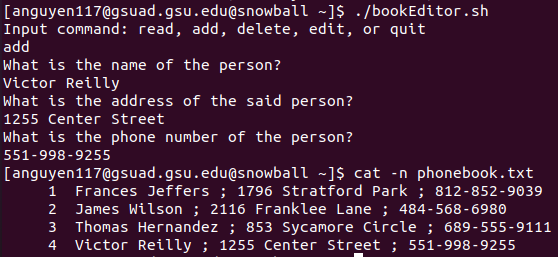
./bookEditor.sh to execute it.

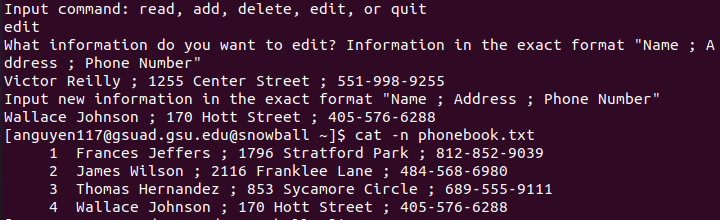
There are five commands which reads the matching input, adds to the phonebook, delete data from the phonebook, edit an existing entry, and simply quit. Type the command to begin.

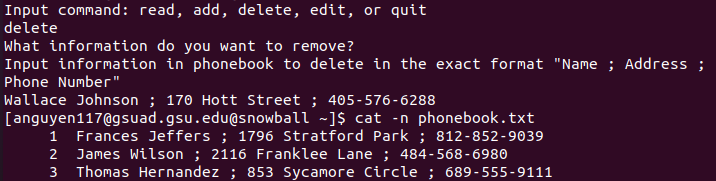












Code for bookEditor.sh (also in uploaded file):

#!/bin/bash

echo "Input command: read, add, delete, edit, or quit"

read input

if [[ "$input" == "add" ]]; then

echo "What is the name of the person?"

read name

echo "What is the address of the said person?"

read address

echo "What is the phone number of the person?"

read telephone

echo "$name ; $address ; $telephone" >> phonebook.txt

elif [[ "$input" == "read" ]]; then

echo "What do you wish to find? Enter part of name, part of address, or part of telephone number"

read info

#match = `grep -i $info phonebook.txt | awk '{print $1}'`

echo " Name ; Address ; number of Phone is as followed: "

grep -i $info phonebook.txt

elif [[ "$input" == "quit" ]]; then

echo "Exiting..."

elif [[ "$input" == "delete" ]]; then

echo "What information do you want to remove?"

echo "Input information in phonebook to delete in the exact format \"Name ; Address ; Phone Number\""

read name1

sed -i "s/$name1//g" phonebook.txt

#sed '/^$/d' phonebook.txt

elif [[ "$input" == "edit" ]]; then

echo "What information do you want to edit? Information in the exact format \"Name ; Address ; Phone Number\""

read name2

echo "Input new information in the exact format \"Name ; Address ; Phone Number\""

read edit

sed -i "s/$name2/$edit/g" phonebook.txt

fi

5.

A. Write a C script that will compute the factorial of a given number (positive integer).

B. Write a C script to find the new integer value of an original integer when it is bit-shifted left by 3 bits and added to its complement (one’s complement of the original integer).

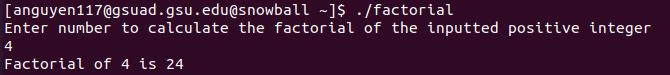
(Note: You can manually type in the binary representation of the original integer)

5a)

Chmod a+x to give permission to execute factorial.c

cc -o factorial factorial.c to compile

./factorial to execute



Code for factorial.c (also in uploaded file):

#include <stdio.h>

int main()

{

int counter, number, factorial = 1;

printf("Enter number to calculate the factorial of the inputted positive integer\n");

scanf("%d", &number);

for (counter = 1; counter <= number; counter++){

factorial = factorial \* counter;

}

if (number < 1)

printf("This is not a valid number.\n");

else

printf("Factorial of %d is %d\n",number,factorial);

return 0;

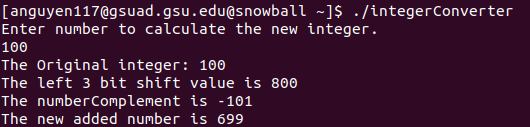
}

5b)

chmod a+x to give permission to execute integerConverter.c

cc -o integerConverter integerConverter.c to compile

./integerConverter to execute



Code for integerConverter.c (also in uploaded file):

#include<stdio.h>

int main()

{

int number =1;

printf("Enter number to calculate the new integer.\n");

scanf("%d", &number);

printf("The Original integer: %d\n",number);

int bitShiftLeft3 = (number) << 3;

//printf(bitShiftLeft3);

int numberComplement = (~number);

//printf(numberComplement);

printf("The left 3 bit shift value is %d \n", bitShiftLeft3);

printf("The numberComplement is %d \n", numberComplement);

int result = bitShiftLeft3 + numberComplement;

printf("The new added number is %d \n", result);

printf("The orignal number is %d \n", number);

return 0;

}

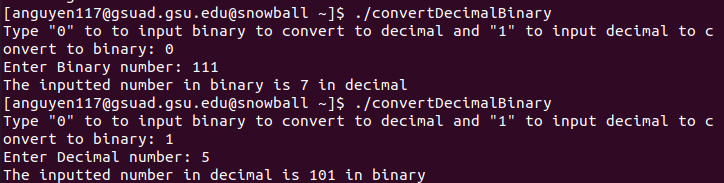
(10 bonus points for writing the C script to convert the integer to binary and vice-versa)

Bonus 1)

chmod a+x to give permission to execute convertDecimalBinary.c

gcc convertDecimalBinary.c -lm -o convertDecimalBinary to compile

./convertDecimalBinary to execute



Code for convertDecimalBinary.c (also in uploaded file):

#include <math.h>

#include <stdio.h>

int main() {

printf("Type \"0\" to to input binary to convert to decimal and \"1\" to input decimal to convert to binary: ");

int input;

scanf("%s", &input);

if (strcmp(&input, "0") == 0){

long long Binary;

printf("Enter Binary number: ");

scanf("%lld", &Binary);

int decima = 0, i = 0, remainder = 0;

while (Binary != 0) {

remainder = Binary % 10;

Binary /= 10;

decima += remainder \* pow(2, i);

++i;

}

printf("The inputted number in binary is %d in decimal\n", decima);

}

else

{

int Decimal;

printf("Enter Decimal number: ");

scanf("%d", &Decimal);

long long binar = 0;

int remainder, n = 1;

while (Decimal != 0){

remainder = Decimal % 2;

Decimal /= 2;

binar += remainder \* n;

n \*= 10;

}

printf("The inputted number in decimal is %lld in binary\n", binar);

}

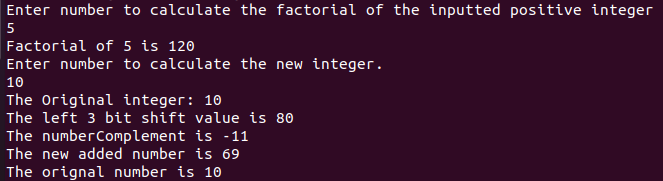
return 0;

}

(10 bonus points for writing a shell script that will execute both the C scripts from above for a given integer number)

chmod a+x doubleExecuterC.sh to give permission to execute doubleExecuterC.sh

./doubleExecuterC.sh to execute



Code for doubleExecuterC.sh (also in uploaded file):

#!/bin/bash

gcc factorial.c -o factorial

./factorial

gcc integerConverter.c -o integerConverter

./integerConverter