

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: Thach Pham

Campus ID: tpham84

Panther #: 002416419

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

```
[tpham84@gsuad.gsu.edu@snowball midterm]$ man sudo | cat >> mandatabase.txt
[tpham84@gsuad.gsu.edu@snowball midterm]$ man ls | cat >> mandatabase.txt
[tpham84@gsuad.gsu.edu@snowball midterm]$ man cat | cat >> mandatabase.txt
[tpham84@gsuad.gsu.edu@snowball midterm]$ man grep | cat >> mandatabase.txt
[tpham84@gsuad.gsu.edu@snowball midterm]$ man vi | cat >> mandatabase.txt
[tpham84@gsuad.gsu.edu@snowball midterm]$ man awk | cat >> mandatabase.txt
[tpham84@gsuad.gsu.edu@snowball midterm]$ man sed | cat >> mandatabase.txt
[tpham84@gsuad.gsu.edu@snowball midterm]$ man chmod | cat >> mandatabase.txt
[tpham84@gsuad.gsu.edu@snowball midterm]$ man wc | cat >> mandatabase.txt
[tpham84@gsuad.gsu.edu@snowball midterm]$ man echo | cat >> mandatabase.txt
[tpham84@gsuad.gsu.edu@snowball midterm]$
```

Copying data into mandatabase.txt file.

```
#!/bin/bash
echo "Type a command"
read command_input
if grep -q $command_input mandatabase.txt; then
    grep -w $command_input mandatabase.txt;
else
    echo "Sorry, I cannot help you"
fi

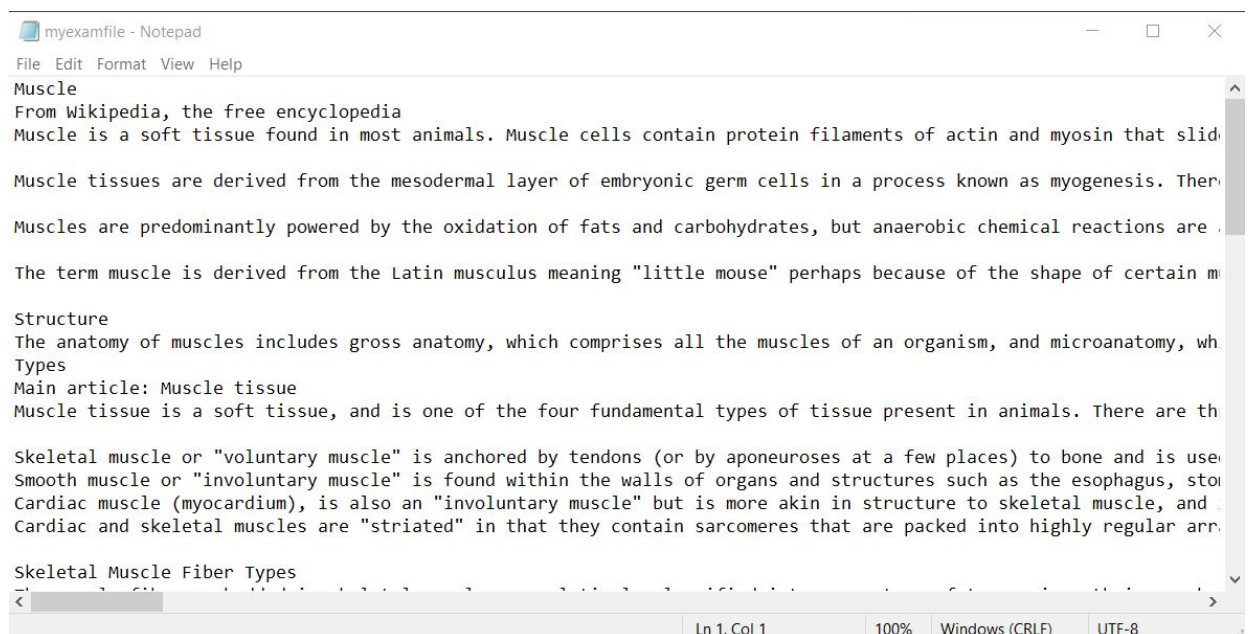
```

helpme.sh script

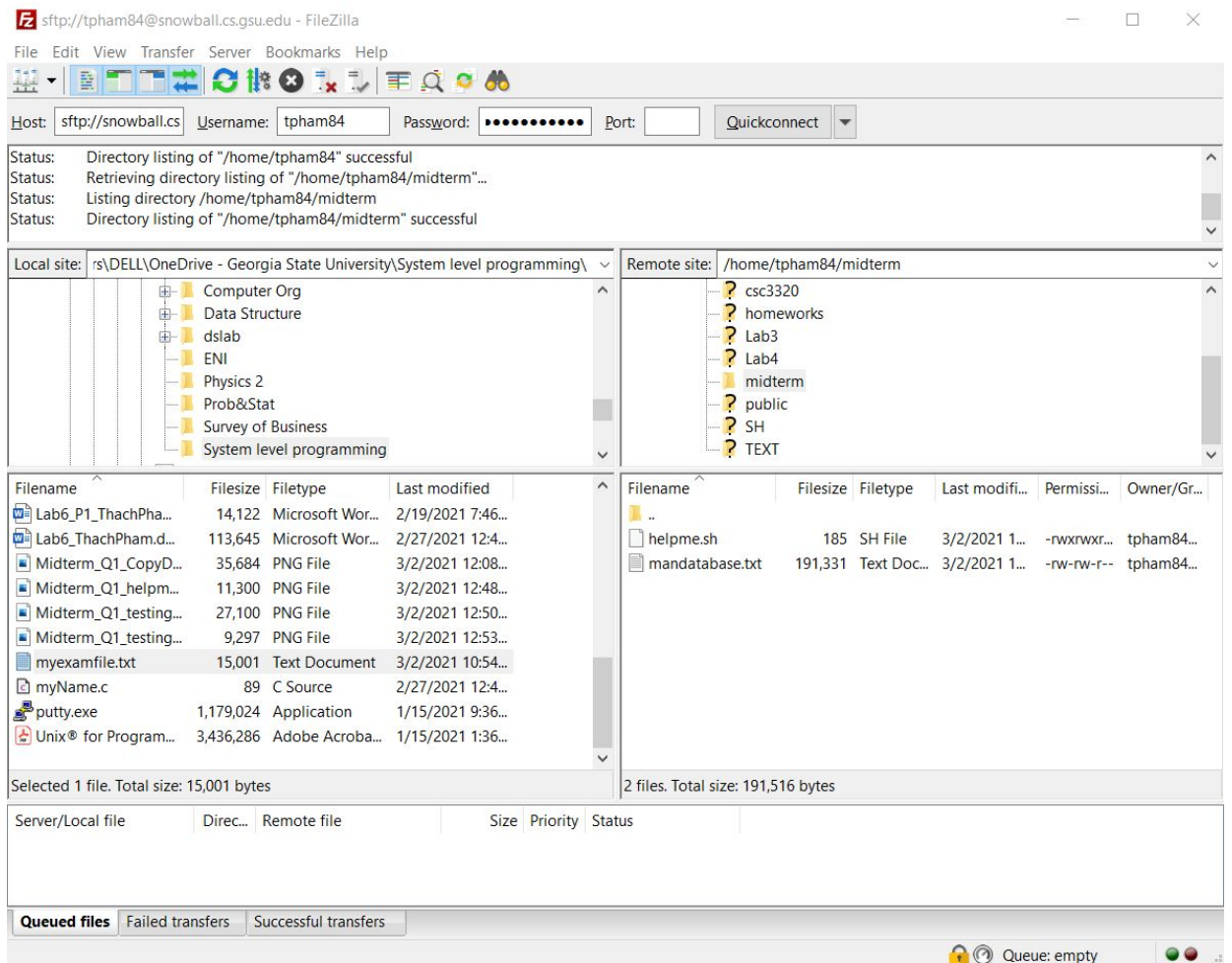
The testing of the script will be attached outside of this document.

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)



Creating myexamfile.txt



Copying the file to the snowball server.

```
tpham84@gsuad.gsu.edu@snowball:~/midterm
#!/bin/bash
#match a keyword provided by the user and find the number of occurrences of that
#word in myexamfile.txt
echo "Type a keyword"
read key_word
grep -o -i $key_word myexamfile.txt | wc -l

```

"matchwordscript.sh" [New] 6L, 199C written 6,1 All

Script to match the keyword and find the number of occurrences. Name of the script is matchwordscript.sh

```
[tpham84@gsuad.gsu.edu@snowball midterm]$ ./matchwordscript.sh
Type a keyword
muscle
135
[tpham84@gsuad.gsu.edu@snowball midterm]$
```

Test 1

```
[tpham84@gsuad.gsu.edu@snowball midterm]$ ./matchwordscript.sh
Type a keyword
fat
8
[tpham84@gsuad.gsu.edu@snowball midterm]$
```

Test 2

```
[tpham84@gsuad.gsu.edu@snowball midterm]$ ./matchwordscript.sh
Type a keyword
fiber
29
[tpham84@gsuad.gsu.edu@snowball midterm]$
```

Test 3

```
[tpham84@gsuad.gsu.edu@snowball midterm]$ ./matchwordscript.sh
Type a keyword
bone
4
[tpham84@gsuad.gsu.edu@snowball midterm]$
```

Test 4

```
[tpham84@gsuad.gsu.edu@snowball midterm]$ ./matchwordscript.sh
Type a keyword
run
2
[tpham84@gsuad.gsu.edu@snowball midterm]$
```

Test 5

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.

```
#!/bin/bash
echo "Enter the number of days"
read num
echo "Enter the name of the compressed file"
read name
find ~/ -iname "*" -atime +$num -type f | zip $name -@
```

Creating the script. The script name is q3script.sh


```

[tpham84@gsuad.gsu.edu@snowball midterm]$ ./q3script.sh
Enter the number of days
5
Enter the name of the compressed file
test1
adding: home/tpham84/csc3320/lab2/myLab2.txt (stored 0%)
adding: home/tpham84/checkError.sh (deflated 43%)
adding: home/tpham84/Lab3/Try.c (deflated 42%)
adding: home/tpham84/Lab3/RealEstate.csv (deflated 77%)
adding: home/tpham84/Lab3/test.txt (stored 0%)
adding: home/tpham84/hello.sh (deflated 32%)
adding: home/tpham84/Lab4/CSC_Course.txt (deflated 75%)
adding: home/tpham84/Lab4/newList.txt (deflated 60%)
adding: home/tpham84/Lab4/mountainList.txt (deflated 60%)
adding: home/tpham84/TEXT.tar.gz (stored 0%)
adding: home/tpham84/csc.sh (stored 0%)
adding: home/tpham84/file2.txt (stored 0%)
adding: home/tpham84/Result (stored 0%)
adding: home/tpham84/homeworks/homework_instructions.txt (deflated 51%)
adding: home/tpham84/simple.sh (deflated 16%)
adding: home/tpham84/TEXT/test_copy.txt (stored 0%)
adding: home/tpham84/TEXT/file3_copy.txt (stored 0%)
adding: home/tpham84/TEXT/file2_copy.txt (stored 0%)
adding: home/tpham84/TEXT/file1_copy.txt (stored 0%)
adding: home/tpham84/.bash_logout (stored 0%)
adding: home/tpham84/abc.sh (stored 0%)
adding: home/tpham84/SH.tar.gz (stored 0%)
adding: home/tpham84/file1.txt (stored 0%)
adding: home/tpham84/test.txt (stored 0%)
adding: home/tpham84/SH/abc_copy.sh (stored 0%)
adding: home/tpham84/SH/csc_copy.sh (stored 0%)
adding: home/tpham84/SH/file_copy.sh (stored 0%)
adding: home/tpham84/SH/simple_copy.sh (deflated 16%)
adding: home/tpham84/SH/SH.tar.gz (deflated 9%)
adding: home/tpham84/public/Submission/Lab2/Lab2_2.txt (deflated 68%)
adding: home/tpham84/public/Submission/Lab2/Lab2_P2/RealEstate.csv (deflated 77%)
adding: home/tpham84/public/myRealEstate.csv (deflated 77%)
adding: home/tpham84/file3.txt (stored 0%)
adding: home/tpham84/file.sh (stored 0%)
[tpham84@gsuad.gsu.edu@snowball midterm]$ ls -ld test1.zip
-rw-rw-r--. 1 tpham84@gsuad.gsu.edu tpham84@gsuad.gsu.edu 69765 Mar  2 00:11 test1.zip
[tpham84@gsuad.gsu.edu@snowball midterm]$ ls
helpme.sh  mandatabase.txt  matchwordscript.sh  myexamfile.txt  q3script.sh  test1.zip
[tpham84@gsuad.gsu.edu@snowball midterm]$ ls -l test1.zip
-rw-rw-r--. 1 tpham84@gsuad.gsu.edu tpham84@gsuad.gsu.edu 69765 Mar  2 00:11 test1.zip
[tpham84@gsuad.gsu.edu@snowball midterm]$ 

```

Output

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.

First we write a script for users to insert name, address and phone numbers to the phone book. The script is named “`addphonebook.sh`” and can be written as follow:

```
#!/bin/bash
#entry purpose for the phone-book utility
echo "Enter name"
read name
echo "Enter address"
read address
echo "Enter phone number(xxx-xxx-xxxx) "
read phone_number
echo "$name ; $address ; $phone_number" >> phone_book.txt
```

Test the script:

```
[tpham84@gsuad.gsu.edu@snowball midterm]$ ./addphonebook.sh
Enter name
Thach Pham
Enter address
46 Ky Hoa
Enter phone number (xxx-xxx-xxxx)
999-999-9999
```

We then write a script to display the phone book in alphabetical order:

```
#!/bin/bash
#Display phone book in alphabetical order
sort -u phone_book.txt
~
```

Test the script:

```
[tpham84@gsuad.gsu.edu@snowball midterm]$ ./displayphonebook.sh
; ;
Aderson James ; 900 Battlelane Circle ; 879-482-8637
Caleb Pog ; 768 Highway Drive ; 080-342-5678
Johnson Baby ; 55 John Street ; 123-456-78910
Thach Pham ; 46 Ky Hoa ; 999-999-9999
Thu Nguyen ; 180 Love Street ; 047-200-2017
```

We then write a script to find an information in phone book, named it "findphonebook.sh":

```
#!/bin/bash
#Find info inside phone book
echo "Enter name, address or phone number to search"
read info
if grep -q $info phone_book.txt; then
    echo "Name ; Address ; Phone Number"
    grep -i $info phone_book.txt;
else
    echo "Sorry, no match found"
fi
```

Test the script:

```
Enter name, address or phone number to search
Thu
Name ; Address ; Phone Number
Thu Nguyen ; 180 Love Street ; 047-200-2017
```

We then write a script to delete an entry in phone book, named it "deletephonebook.sh":

```
#!/bin/bash
#Delete an entry in phone book
echo "Enter a name to be deleted"
read name
sed -i "/$name/d" phone_book.txt
```

Test the script:

```
[tpham84@gsuad.gsu.edu@snowball midterm]$ ./displayphonebook.sh
; ;
Aderson James ; 900 Battlelane Circle ; 879-482-8637
Caleb Pog ; 768 Highway Drive ; 080-342-5678
Johnson Baby ; 55 John Street ; 123-456-78910
Thach Pham ; 46 Ky Hoa ; 999-999-9999
Thu Nguyen ; 180 Love Street ; 047-200-2017
[tpham84@gsuad.gsu.edu@snowball midterm]$ vi deletetphonebook.sh
[tpham84@gsuad.gsu.edu@snowball midterm]$ ./deletetphonebook.sh
Enter a name to be deleted
Caleb
[tpham84@gsuad.gsu.edu@snowball midterm]$ ./displayphonebook.sh
; ;
Aderson James ; 900 Battlelane Circle ; 879-482-8637
Johnson Baby ; 55 John Street ; 123-456-78910
Thach Pham ; 46 Ky Hoa ; 999-999-9999
Thu Nguyen ; 180 Love Street ; 047-200-2017
```

We now combine all the scripts into one master script that can prompt for user input. The script will be called "phone_book_utiliy.sh":


```
#!/bin/bash
#Master script for all scripts
exit=0
while [ $exit -ne 1 ]
do
echo "add, find, delete, display, exit:"
read input
if [ "$input" = "add" ]
then
bash /home/tpham84/midterm/addphonebook.sh
elif [ "$input" = "find" ]
then
bash /home/tpham84/midterm/findphonebook.sh
elif [ "$input" = "display" ]
then
bash /home/tpham84/midterm/displayphonebook.sh
elif [ "$input" = "delete" ]
then
bash /home/tpham84/midterm/deletetphonebook.sh
elif [ "$input" = "exit" ]
then
exit=1
fi
done
exit 0
```

We test all the function of the script:

```

[tpham84@gsuad.gsu.edu@snowball midterm]$ ./phone_book_utility.sh
add, find, delete, display, exit:
add
Enter name
Pham Huy
Enter address
860 Kibbe Circle
Enter phone number(XXX-XXX-XXXX)
470-401-9701
add, find, delete, display, exit:
display
Aderson James ; 900 Battlelane Cir ; 879-482-8637
Aderson James ; 900 Battlelane Circle ; 879-482-8637
Johnson Baby ; 55 John Street ; 123-456-78910
Pham Huy ; 860 Kibbe Circle ; 470-401-9701
Thu Nguyen ; 180 Love Street ; 047-200-2017
add, find, delete, display, exit:
find
Enter name, address or phone number to search
Thach
Sorry, no match found
add, find, delete, display, exit:
find
Enter name, address or phone number to search
Thu
Name ; Address ; Phone Number
Thu Nguyen ; 180 Love Street ; 047-200-2017
add, find, delete, display, exit:
delete
Enter a name to be deleted
Pham
add, find, delete, display, exit:
display
Aderson James ; 900 Battlelane Cir ; 879-482-8637
Aderson James ; 900 Battlelane Circle ; 879-482-8637
Johnson Baby ; 55 John Street ; 123-456-78910
Thu Nguyen ; 180 Love Street ; 047-200-2017
add, find, delete, display, exit:
exit
[tpham84@gsuad.gsu.edu@snowball midterm]$ █

```

Lastly, we change the permission of the phone_book.txt file so only the user has access to it.

```
[tpham84@gsuad.gsu.edu@snowball midterm]$ chmod 700 phone_book.txt
[tpham84@gsuad.gsu.edu@snowball midterm]$ ls -ld phone_book.txt
-rwx-----. 1 tpham84@gsuad.gsu.edu tpham84@gsuad.gsu.edu 193 Mar  2 05:03 phone_book.txt
[tpham84@gsuad.gsu.edu@snowball midterm]$
```

5.

A. Write a C script that will compute the factorial of a given number (positive integer). Script is attached below:

```
#include <stdio.h>
int factorial(int num){
if(num==0) return 1;
return num*factorial(num-1);
}
int main(){
int num;
printf("Enter a value that you want to find the factorial: \n");
scanf("%d", &num);
printf("Factorial: %d \n", factorial(num));
return 0;
}
```

~~~~~



Test the script:

```
[tpham84@gsuad.gsu.edu@snowball midterm]$ cc -o intfactorial intfactorial.c
[tpham84@gsuad.gsu.edu@snowball midterm]$ ./intfactorial
Enter a value that you want to find the factorial:
5
Factorial: 120
```

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)

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

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

We write the script doing so and named it "binaryoperation.c" (sorry for the typo I made when naming the file). The script to convert integer to binary and vice versa is also included in this script:

```

#include <stdio.h>
#include <math.h>
int convertBinaryToInteger(int binaryNum){
int decimal=0; int i=0; int remainder;
while (binaryNum != 0){
remainder= binaryNum % 10;
binaryNum /= 10;
int power = pow(2,i);
decimal += remainder*power;
++i;
}
return decimal;
}

long long convertIntegerToBinary(int integerNum){
long long binary=0;
int remainder; int i=1;
while(integerNum!=0){
remainder = integerNum%2;
integerNum /= 2;
binary += remainder*i;
i *= 10;
}
return binary;
}

int isBinary(int number){
int last;
while (number){
last=number%10;
if (last!=0 && last!=1){ //Find if the last digit is different than 0 and 1 because binary only consists of 0 and 1
return 0;
}
number = number/10;
}
return 1;
}

int operation(int num){
int shift3 = num<< 3;
int complement = ~(num);
return shift3+complement;
}

int main(){
printf("Enter a number you want to do the operation on, can be binary or decimal \n");
int num;
scanf("%d",&num);
if(isBinary(num)){ //check if the number entered by user is binary
"binaryoperation.c" 56L, 1190C

```

```

int main()
printf("Enter a number you want to do the operation on, can be binary or decimal \n");
int num;
scanf("%d",&num);
if(isBinary(num)){ //check if the number entered by user is binary
long long binary = (long long) num;
num = convertBinaryToInteger(binary);
}
int new_num = operation(num);
printf("New number is: %d \n In binary: %lld", new_num, convertIntegerToBinary(new_num));
return 0;
}

```

We test the script:

```
Enter a number you want to do the operation on, can be binary or decimal
5
New number is: 34
In binary: 100010[tpham84@gsuad.gsu.edu@snowball midterm]$ vi binaryoperaion.c
```

Now we create a shell script that runs both script in part A and part B when called:

```
#!/bin/bash
```

```
cc -o intfactorial intfactorial.c
```

```
./intfactorial
```

```
cc -o binaryoperation binaryoperation.c -lm
```

```
./binaryoperation
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```