CSc 3320: Systems Programming

Spring 2021

Final/Project: Total points = 100

THIS FINAL IS OPTIONAL

Assigned: 23th Apr 2021, Friday Noon

Submission Deadline (if attempting): 2nd May 2021, Sunday, 11.59 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: Rafid H. Shaon

Campus ID: rshaon1

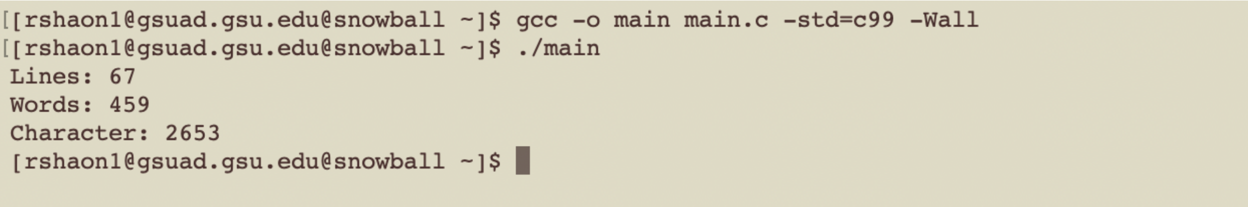
Panther #: 002-49-7367

All programs have to be well commented. Non commented programs will receive 0 points. Comments have to be easily comprehensible and concise.

1. [30pts] Copy the contents of this document into a text file. Make sure the spacings and indentations are included.
   Write a C program that reads the text file and then outputs
   -- the number of characters (space is to be considered a character),
   -- number of words (a word is any sequence of non-white-space characters),
   -- number of lines.
   Write a makefile as well.

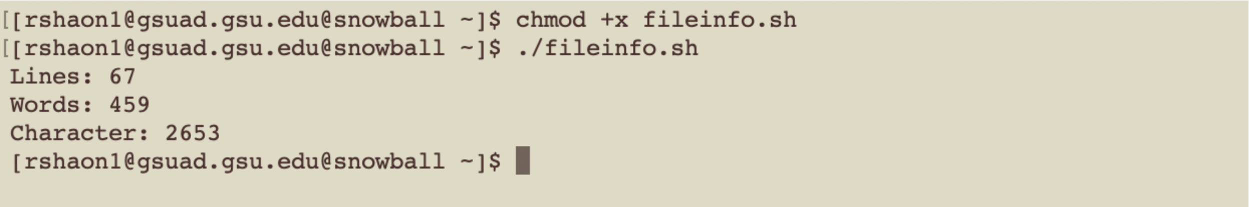
   Check text.txt for text file.

   Check Makefile for makefile.



```
[[rshaon1@gsuad.gsu.edu@snowball ~]$ gcc -o main main.c -std=c99 -Wall
[[rshaon1@gsuad.gsu.edu@snowball ~]$ ./main
 Lines: 67
 Words: 459
 Character: 2653
 [rshaon1@gsuad.gsu.edu@snowball ~]$ 
```

2. Repeat question 1, but write a shell script instead of C. Makefile not necessary. [30pts]



```
[[rshaon1@gsuad.gsu.edu@snowball ~]$ chmod +x fileinfo.sh
[[rshaon1@gsuad.gsu.edu@snowball ~]$ ./fileinfo.sh
 Lines: 67
 Words: 459
 Character: 2653
 [rshaon1@gsuad.gsu.edu@snowball ~]$ 
```

3. [40pts] Describe (briefly in 1-2 sentences) the following unix utility functions and provide 1 example of it's usage. You can refer to Chapter 13 in the Unix textbook. You must NOT provide the same example from the textbook:
    a. perror()
    b. open()
    c. read()
    d. write()
    e. lseek()
    f. close()
    g. monitor()
    h. chown()
    i. fchown()
    j. chmod()
    k. fchmod()
    l. link()
    m. unlink()
    n. getpid()
    o. getppid()
    p. fork()
    q. exit()
    r. wait()
    s. alarm()
    t. signal()
    u. kill()
    v. pipe()
    w. scp() (also referred to as secure copy)

Check either ex.txt or ex3 doc. for all answers in question #3.