**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: Tracy Michaels

Campus ID: tmichaels1

Panther #: 002430918

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

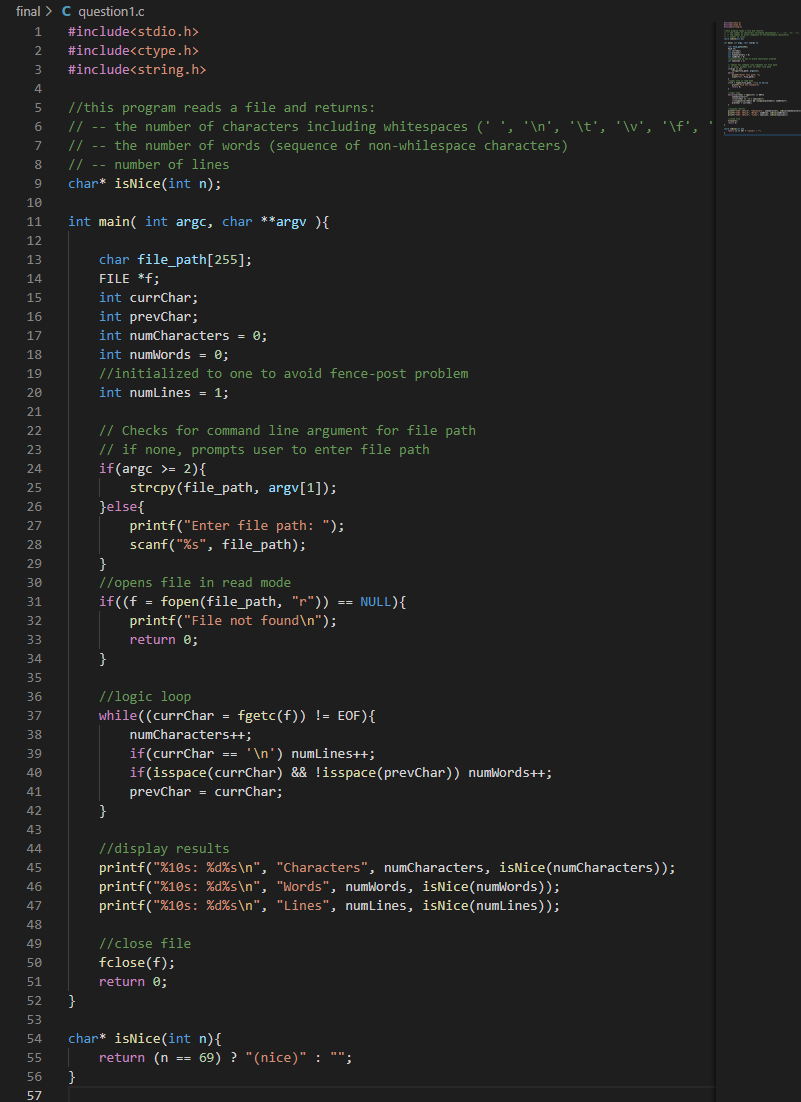
1. [30pts] Copy the contents of this document into a text file. Make sure the spacings and indentations are included.

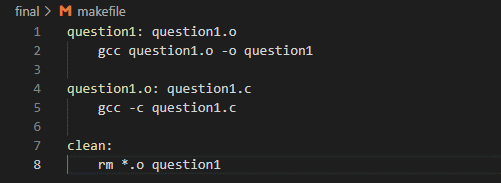
Write a C program that reads the text file and then outputs

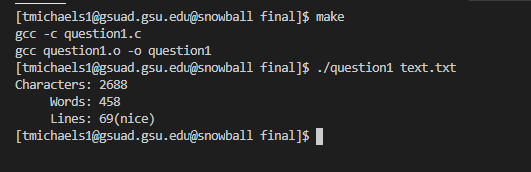
-- the number of characters (space is to be considered a character),

-- number of words (a word is any sequence of non-white-space characters), -- number of lines.

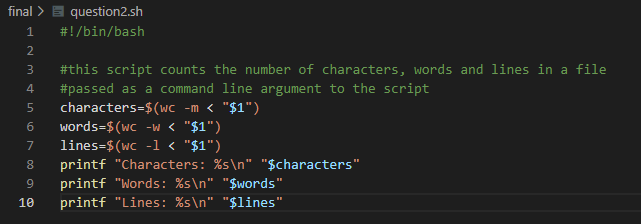
Write a makefile as well.

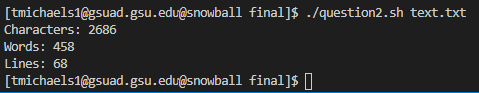






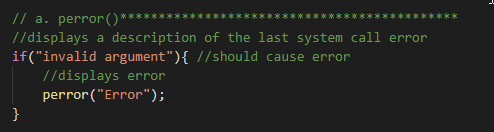
1. Repeat question 1, but write a shell script instead of C. Makefile not necessary. [30pts]





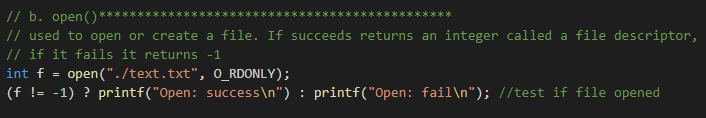
1. [40pts] Describe (briefly in 1-2 sentences) the following unix utility functions and provide 1 example of it’s usage. You can refer to Chapter 13 in the Unix textbook. You must NOT provide the same example from the textbook:
   1. perror()

* displays a description of the last system call error



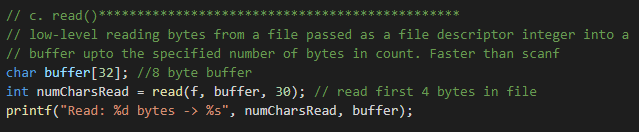


* 1. open()



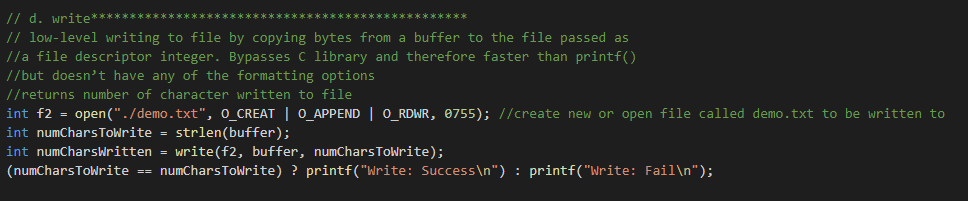


* 1. read()





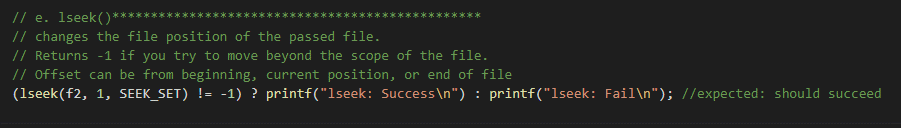
* 1. write()







* 1. lseek()

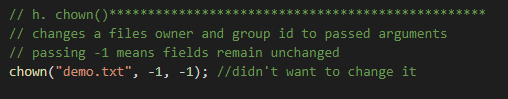




* 1. close()
* closes the files that was opened by freeing the file descriptor

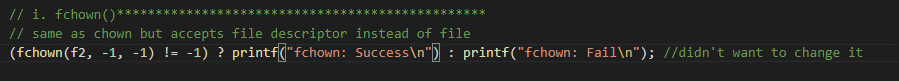


* 1. monitor()
* perodically scans and displays information about a file
* $ monitor text.txt
  1. chown()



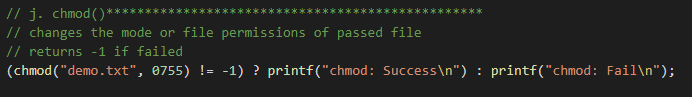


* 1. fchown()



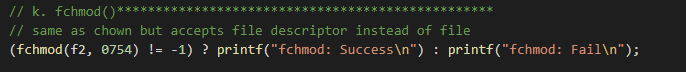


* 1. chmod()



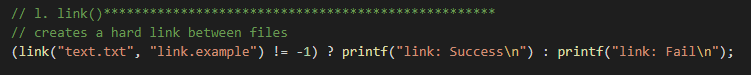


* 1. fchmod()

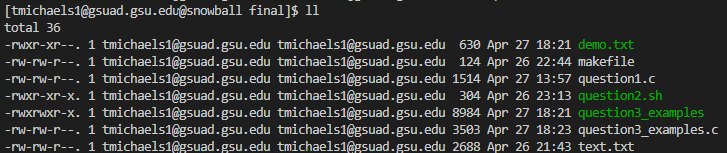




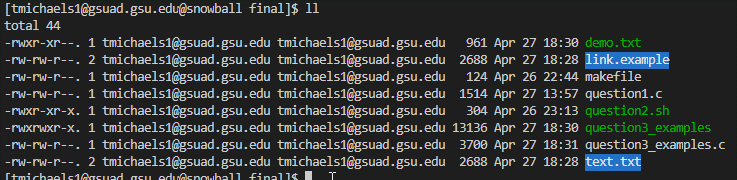
* 1. link()



Before:

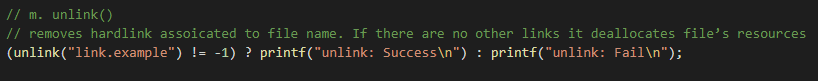


After:

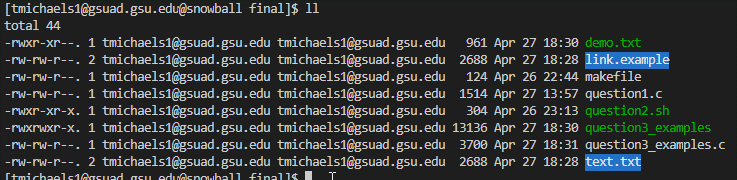




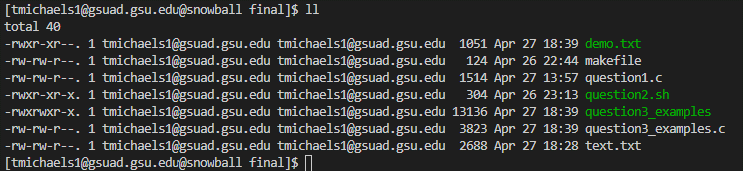
* 1. unlink()



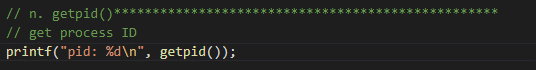
before:



After:

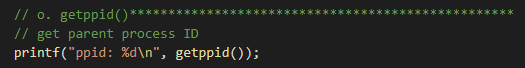


* 1. getpid()



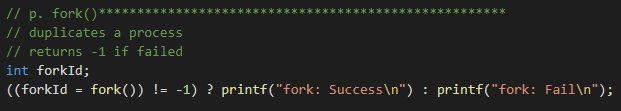


* 1. getppid()



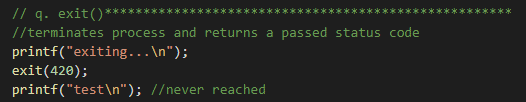


* 1. fork()



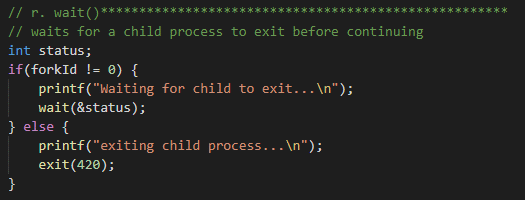


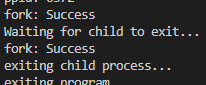
* 1. exit()



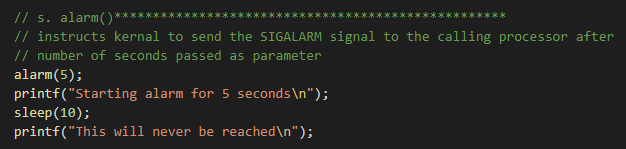


* 1. wait()





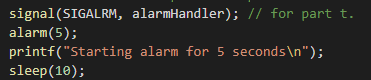
* 1. alarm()



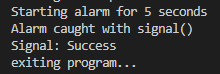


* 1. signal()

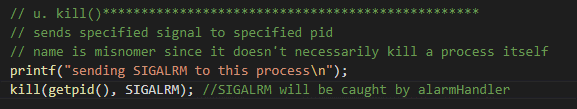






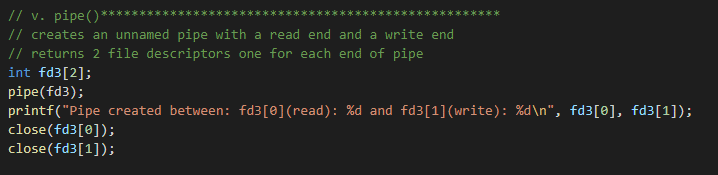


* 1. kill()





* 1. pipe()





* 1. scp() (also referred to as secure copy)
* stands for secure copy protocol
* securely copies files between hosts on a network
* ex: $ scp text.txt text\_copy.txt (this example copies inplace, but with the right flags can be done over network)





*Entirety*