CS3.304 - Advanced Operating Systems
Assignment - 1
Deadline : August 12, 2023 11.59PM

Q1:

- Given a file, you need to reverse the contents of the file and store the result in a new file in the directory named "**Assignment1_1**".
- The percentage of the file written should be printed on the console during file writing.
- The percentage of the file written should be overwritten each time (shouldn't write multiple times).
- The directory created should have read, write, and execute permissions for the user who created it.
- The new file created should have the read and write permissions for the user who created it.
- The program will be tested on LARGE (>1GB) files which could be greater than RAM size.

- The input file path would be given to you as an argument to your program :
  $ gcc Q1.c / g++ Q1.cpp
  $ ./a.out <input file name>

- The output file must be named:
  1_<input file name>

- **Example:**
  Input: 'A.txt' → "My name is abcd"
  Output: 'Assignment1_1/1 A.txt' → "dcba si eman yM"

Q2:

- Write a program to reverse a specific portion of the file and store the result in a new file in the directory named "**Assignment1_2**".
- The percentage of the file written should be printed on the console during file writing.
- The percentage of the file written should be overwritten each time (shouldn't write multiple times).
- The directory created should have read, write, and execute permissions for the user who created it.

- The new file created should have the read and write permissions for the user who created it.
- The program will be tested on LARGE (>1GB) files which could be greater than RAM size.
- The input file path would be given to you as an argument to your program along with the start and stop character, reverse the rest of the file and leave the portion of the file from the start character to the stop character untouched. In order to do this you will have to reverse the file in two parts, from 0 to the start character and from the end character to the end of the document. Assume the first character of the document to be 0:

  $ gcc Q2.c / g++ Q2.cpp
  $ ./a.out <input file name> <start_index> <end_index>

- The output file must be named:
  2_<input file name>

- **Example**:
  Input:
  ./a.out A.txt 4 7
  'A.txt' → "My name is abcd "
  Output: 'Assignment1_2/2 A.txt' → " n yMame dcba si "

  Input :
  ./a.out B.txt 4 8
  'B.txt' → "0123456789abc"
  Output: 'Assignment1_2/2 B.txt' → "321045678cba9"

Q3:
- Write a program to :
    1. Check the permissions for the two files and the directory.
    2. Check whether the content in the new file are the reverse of the old file.
- The input to this program has paths for newfile, oldfile, and the directory created in Q1.

- **Input:**
  Path of newfile oldfile and directory will be passed as a command-line argument.
  ./a.out <newfile> <oldfile> <directory>

- **Output:**
  Directory is created: Yes

Whether file contents are reversed in newfile: Yes
User has read permissions on newfile: Yes
User has write permission on newfile: Yes
User has execute permission on newfile: No
Group has read permissions on newfile: No
Group has write permission on newfile: No
Group has execute permission on newfile: No
Others has read permissions on newfile: No
Others has write permission on newfile: No
Others has execute permission on newfile: No

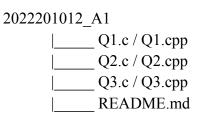- The above 9 lines should be printed for the old file and the directory too.

Generation of Random String File :

- To generate the random string file, a python script is uploaded along with this pdf. It contains all the instructions required to run it.

Guidelines:

1. Assignment should be coded in C/C++.
2. All Programs must use system calls only.
3. Useful commands: read, write, lseek, stat, fflush, perror.
4. Use man pages exclusively.
5. Use of system commands like ls, cp, mv, mkdir, etc are not allowed. You have to implement your own versions of these commands using system calls.
6. Modularize and Indent your codes. Also add comments wherever necessary to promote readability.
7. Handle error cases wherever required.
8. Add a README.md File (compulsory) which contains instruction to execute your code and working procedure of your code.
9. Submission by email/teams to TAs will not be accepted.
10. **ZERO tolerance towards any kind of plagiarism. DO NOT copy or share code/code snippets (even a few lines of copied code would be detected and punished) - both the parties will get zero.**

Submission Format:

```
2022201012_A1
        |_____ Q1.c / Q1.cpp
        |_____ Q2.c / Q2.cpp
        |_____ Q3.c / Q3.cpp
        |_____ README.md
```

Follow the above mentioned directory structure and zip the RollNo_A1 folder and submit RollNo_A1.zip on moodle.

Note: All submissions which are not in the specified format or submitted after the deadline will be awarded 0 in the assignment.