

CS3.304 - Advanced Operating Systems

Assignment - 1

Deadline: August 16, 2024 11.59 PM

Q1:

- Given a file, you need to reverse the file's contents and store the result in a new file in the directory named "**Assignment1**".
- 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 (it shouldn't be written 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.
- If the flag is set to 0, the entire file will be reversed. If the flag is set to 1, the start and end indices must be provided as command-line arguments. In this case, the file will be reversed in two parts: from the beginning to the start index, and from the end index to the end of the document, leaving the portion of the file between the start and end indices unchanged. Assume the first character of the document is indexed as 0.

- 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> <flag> (either 0 or 1) <start_index> <end_index>
```

- The output file must be named "0_<input file name>" if the input flag is 0, and "1_<input file name>" if the input flag is 1.

- **Example for flag 0:**

Input: ./a.out A.txt 0 → "Hello, World!"

Output: 'Assignment1/0_A.txt' → "!dlroW ,olleH"

- **Example for flag 1:**

Input :

./a.out A.txt 1 2 7

'A.txt' → "Hello, World!"

Output: 'Assignment1/1_A.txt' → "eHllo, W!dlro"

Input :

./a.out B.txt 1 4 8

'B.txt' → "0123456789abc"

Output: 'Assignment1/1_B.txt' → "321045678cba9"

Input :

./a.out C.txt 1 9 25

'C.txt' → "The quick brown fox jumps over the lazy dog"

Output: 'Assignment1/1_C.txt' → "kciuq ehT brown fox jumps god yzal eht revo"

Q2 :

- 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:**

The path of newfile, oldfile, and directory will be passed as a command-line argument.
./a.out <newfile_path> <oldfile_path> <directory_path>

- **Output:**

Directory is created: Yes

Whether file contents are reversed in newfile: Yes

Both Files Sizes are Same : 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 last 9 lines should be printed for the old file and the directory too. Total 30 lines (3 + 9 + 9 + 9).

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++ only.
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 instructions to execute your code and the working procedure of your code.
9. Submission by email/teams to TAs will not be accepted.

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 parties will get zero.

Submission Format:

2023201012_A1

|_____ 2023201012_A1_Q1.c / 2023201012_A1_Q1.cpp

|_____ 2023201012_A1_Q2.c / 2023201012_A1_Q2.cpp

|_____ README.md

Please replace "2023201012" with your Roll Number.

Follow the specified directory structure and create a zip file named "RollNo_A1.zip" containing the "RollNo_A1" folder. Submit this zip file on Moodle.

Note: Any submissions that do not adhere to the specified format or are submitted after the deadline will receive a score of 0 for the assignment. This deadline is strict and non-negotiable.