



Assignment: 1

Intro to System Calls

Deadline : 18th August 2022, 10:00 AM

1. Part 1:

1. Task:

Given a file, you need to reverse the contents of the file and store the result in a new file in the directory named "Assignment".

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:

```
./a.out < input_file_path >
```

The output file must be named:

```
1_ < input_file_name >
```

Example:

Input: 'A.txt' → "My name is abcd"

Output: 'Assignment/1_A.txt' → "dcba si eman yM"

2. Part 2

1. Task:

Write a program to reverse a specific portion of the file and store the result in a new file in the directory named "Assignment".

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:

```
./a.out < input_file_path > start_char end_char
```

The output file must be named:

```
2_ < input_file_name >
```

Example:

Input:

```
./a.out A.txt 3 7
```

'A.txt' → "My_name_is_abcd_"

Output: 'Assignment/2_A.txt' → "_yMname__dcba_si_"

Input :

```
./a.out B.txt 4 8
```

'B.txt' → "0123456789abc"

Output: 'Assignment/2_B.txt' → "321045678cba9"

3. Part 3:

1. Task: 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.

Example:

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 should be printed for the old file and the directory too.
Path of newfile oldfile and directory will be passed as a command-line argument.

```
./a.out < newfile >< oldfile >< directory >
```

4. Guidelines:

1. All Programs must use system calls only. The use of printf and scanf are restricted
Use of sprintf is only allowed for formatting strings and printing progress
Use of string.h library is permitted
2. Useful commands: read, write, lseek, stat, fflush, perror
3. Use [man pages](#) exclusively.
4. Assignment should be coded in C. Indent your codes and add comments wherever necessary to promote readability
5. For error handling, please use the “perror” command wherever possible
6. Add a README.md File (compulsory) which contains any assumptions made.
7. Submission_format: <RollNo>_Assignment1.tar.gz
8. Submission by email to TAs will not be accepted
9. **Plagiarism will lead to serious consequences.**