Project 4

100 Possible Points

12/5/2023

Add Comment

12/8/2023

Details

Problem Statement

You are to implement functions to seek within a file, delete a file, write to a file, and copy a file within the context of a trivial file system.

Discussion

You will find details about the trivial file system (tfs) in the header file **tfs.h (https://clemson.instructure.com/courses/204138/files/18706844?wrap=1)** **(https://clemson.instructure.com/courses/204138/files/18706844/download?download_frd=1)** , which is available along with related source files, test drivers, and expected output files in the Project 4 subdirectory on Canvas.

The file tfs_helper.c is the first part of the implementation of the tfs file system and contains nine helper functions. The file tfs_public_1.c is the second part of the implementation and contains nine public method functions. The file tfs_public_2.c is a skeleton file in which you will implement four functions to complete the remaining part of the tfs file_system: tfs_seek(), tfs_delete(), tfs_write(), and tfs_copy().

tfs_public_2.c contains header comments for each of these functions, and these comments specify the operations that each function must provide and the constraints and preconditions under which each must operate. In particular, note that tfs_write() must use tfs_block_read() and tfs_block_write() to access bytes of storage and that tfs_copy() must use tfs_block_read() and tfs_block_write(), or tfs_read() and tfs_write(), to access bytes of storage.

You will submit your completed version of tfs_public_2.c file using Gradescope for autograding. You are not allowed to submit any other files.

Within your own programming environment, you may modify the supplied source files and add debugging information to help you in developing your four functions. For example, these files already have options to turn on function call logging and error logging. **However, remember that for grading your tfs_public_2.c file will be compiled with the original versions of the header and other**

supplied source files. So, for instance, you cannot make changes in tfs.h and expect those changes to be available during the Gradescope testing. Also, any debugging print statements you add within tfs_public_2.c should be disabled for grading.

You can compile your code with one of the test drivers and the other function implementation files in the following manner:

```
gcc -Wall tfs_driver_1.c tfs_helper.c tfs_public_1.c tfs_public_2.c
```

Pedagogical Rationale

This assignment reinforces file management concepts including directories, file structures using a file allocation table, and file pointers.

Guidelines

The code should be written totally by yourself.

You may discuss the project requirements and the concepts with me or with anyone in the class.

However, you should not send code to anyone or receive code from anyone, whether by email, printed listings, photos, visual display on a computer/laptop/cell-phone/etc. screen, or any other method of communication.

Do not post the assignment, or a request for help, or your code on any web sites.

The key idea is that you shouldn't short-circuit the learning process for others once you know the answer. (And you shouldn't burden anyone else with inappropriate requests for code or "answers" and thus short-circuit your own learning process.)

Grading

There are four test cases for which the output must match exactly. The test drivers and the expected outputs are available on Canvas in the Project 4 subdirectory. The test cases will be equally weighted.

Note that a corrupted submission or a submission that does not compile will receive 0 points.

Note that direct access to the storage[] array or blocks[] arrays in the tfs_write() and tfs_copy() functions - or equivalent ways of avoiding the proper use of tfs_block_read() and tfs_block_write() and/or tfs_read() and tfs_write() - is disallowed. **The penalty for bypassing tfs_block_read() and tfs_block_write() or tfs_read() and tfs_write() will be up to 80% of the project grade.**

Also, please note that similarity checking will be used to help detect plagiarism. The Clemson policy on Academic Integrity is available **here (https://catalog.clemson.edu/content.php?catoid=39&navoid=1230#undergraduate-academic-integrity)** .