Project: File System

Protocol for Submission and Testing

Documentation

- Submit your source code to the appropriate bin on IVLE by the due date
- The code you submit must be *exactly* the code you use for your testing. If you need to make any changes after submission then inform the lab assistant prior to testing.
- No other documentation is necessary.

Testing

- You need to see the lab assistant during the first lab session after the project is due (see course page for dates).
- You should bring your own laptop for the test. If you do not have a laptop or prefer not to use it, then you can run your program on one of the lab computers.
- Your program must be able to read a text file (.txt extension, similar to the one posted on the web) from a USB memory stick, and write a text file (similar to the one posted on the web) to the same memory stick.
- You will be asked to perform the following steps
    - start your program
    - your program should read the text file called input.txt from a memory stick that will be given to you
    - it should write all output to a single text file on the same memory stick; the name of the file should be nnn.txt, where nnn is your matriculation number
- You only get one chance to run the test, except when there is some minor problem that results in a catastrophic failure and can be fixed on the spot, e.g., the program crashes and produces no results.
- We will evaluate the output of your program and report the results to you (not during the demo session). You can approach the lab assistant after the results have been posted to see the test cases you have failed. If you have a valid justification for why your results are different, we may accept the results or award additional credit.
- I suggest that you test the protocol before coming to the demo session to avoid unnecessary delays/problems:
    - copy the sample input file from the web page onto a memory stick
    - run the above protocol
    - check your memory stick to make sure it contains a file nnn.txt that matches the output file on the web page

Expectations

- There will be only one input file. It contains a series of tests, each starting with an “in” command. It may also contain blank lines. For each blank line in the input file, generate a blank line in the output file (for visual separation).
- All commands in the input file will have a correct format. Specifically, only existing opcodes will be given (cr, op, rd, etc.) followed by the appropriate number of parameters, each separated by a blank space (as shown in the sample input file).
- However, commands may request operations that are illegal, such as closing a file that is not open, creating multiple files with the same name, opening an already open file, etc. It is part of your assignment to identify and catch all such illegal operations. If any such a command is detected, simply write “error” into the output file and continue with the next command (see the sample output).
- The output file should contain a separate line for each input command. The exact input and output formats are given in “Shell Specification” on the course page. To facilitate grading, do not output any additional information.