CS352: Operating System

Prof. Shervin Shirmohammadi

Project: Portable File Manipulator (pofm)

Note: this project will be done in groups of 4 people.

Details

Design a portable file manipulator, called *pofm*, which will operate on files. *pofm* must have the following attributes:

- It must be written in the C programming language.
- Since it must be portable, it must NOT use any OS commands, so that *pofm* is easily ported to any operating system and platform. All commands in your program must be standard C functions, and you cannot use any system() calls.
- *pofm* must provide the following functionality:
  - Create a new file
  - Delete a file
  - Rename a file
  - Copy a file
  - Move a file from one directory to another
  - For text files only:
    - Append text to the end of a file
    - Insert text in a specific position (counted in characters) of the file
    - Remove all text in a file
    - Show the content of a text file, with the ability to pause per page. The number of lines per page can be specified by the user.
- All commands must have a "help" utility; i.e., if the user types command1 /h, for example, then the program must show how the command is used.
- All commands must catch exceptions and problems; e.g., file does not exist, file is read-only, file name already exists, invalid command usage, etc.
- You can name your commands as you like.

Deliverables and Marking Scheme:

1. In-class **demo (40%)**. The TA will come to your groups, one by one, will see your *pofm* running, and will ask you to show specific features and also will ask you questions about your system and about your code.

Please email the TA to fix your demo date (first come, first serve). Please mention in your email to the TA who are the group members:
May 3 (3 demos)
May 8 (7 demos)
May 10 (3 demos)

2. **Final report (60%)**: no later than May 12. This report consists of 3 components:
   a) A design guide (30%): start by a short introduction, followed by explanation of your design and implementation. Give reasons for your design and implementation choices. There should be enough details in your report to explain everything. I do not set number of pages: it's up to you and how many pages you need to explain things.
   b) A user manual including examples and screenshots (20%).
   c) All your code, with in-line comments (10%).