

Data Structures and Algorithms

Project (Linked List, Vectors, and Generalized Trees)

Deadline: 18/12/2023

It has two parts

Part 1 - You need to implement a DOS shell.

You need to implement a dos/command window like a shell. Use the command window of your operating system to understand its working.

When your program starts its execution, the program shows the author's name, registration number, and course name at the top line. Below this information the user is shown to be in a virtual drive V:\> and waits for input. The input commands and their interpretation are given below:

ATTRIB	Displays file (name provided as input) attributes.
CD	Displays the name of or changes the current directory. See details below.
CD.	Prints working directory (i.e. your current directory in your tree structure)
CD..	Change directory to previous directory
CD\	Changes directory to root directory (i.e. V:\>)
CONVERT	Asks two types and converts extension of all files of one type to another type.
COPY	Copies one file in the current directory to another location (directory). See details below.
CREATE	Creates a file with the name provided and allows the user to enter contents of the file.
DEL	Delete a file whose name is provided in input.
DIR	Displays a list of files and subdirectories in a directory.
EDIT	Opens a file (loads into a linked list) and allows the user to edit and save the contents of the file. See bullet 5 below
EXIT	Quits the program
FIND	Searches for a file in your current virtual directory whose name is provided as input.
FINDF	Searches for a text string in the currently open file or the file whose name is provided as input.
FINDSTR	Searches for strings in all files of your current virtual directory, prints names of files with the string
FORMAT	Formats the current virtual directory i.e. empties the current directory and all subdirectories.
HELP	Provides Help information for all these commands.
LOADTREE	Load a given tree in your tree structure. Given tree is in a file named tree.txt
MKDIR	Creates a virtual directory.
MOVE	Moves one file (whose name is provided as input) from one directory to another directory.
PPRINT	Adds a text file to the priority based print queue, and displays the current priority queue.
PROMPT	Changes the Windows command prompt (for example from V:\> to V\$).

This assignment/Project is mainly contributed by Dr. Zeeshan Rana and in collaboration with Dr. Atif Mehdi. Thanks a lot for their help.

PRINT	Adds a text file to the print queue, and displays the current queue.
PQUEUE element	Shows current state of the priority based print queue, with time left for each element
PWD	Prints working directory
QUEUE	Shows current state of the print queue, with time left for each element
RENAME	Renames a file whose current and new name is provided as input.
RMDIR	Removes a directory along with its contents. Consider setting the respective sibling links
SAVE	Saves the currently open file to disk.
TREE	Displays the complete directory structure.
VER	Displays the version of your program.

Note:

1. Assume you can have text files only
2. Directories are not actual directories of your computer. These directories are tree nodes in your tree structure.
3. There can be multiple directories (and/or files) in a directory
4. **One node in your directory structure may be a file or a directory.** Each file has a file name, path, number of characters, number of lines, average number of characters per line, creation time, owner, readOnly property, priority, time it takes to print, and file type; the allowed types are .txt and .text only. Like a file, a directory also has a directory name, creation time, owner, readOnly property etc. Consider a file and a directory having the same type. Attributes like priority, readOnly property, and time to print may be requested as input while saving a file.
5. **CREAT, EDIT, and FIND:**
Text files (in current virtual folder i.e. according to your tree hierarchy) should be loaded in linked lists (**STL MUST be used to avoid hassle of creating and managing the lists**) when editing and searching. Do not use built in functions (except **GotoRowCol(ri, ci)**) and libraries for searching and file editing. The edited linked list may then be written to disk using built in support for filing.
6. **TREE COMMAND:**
You need to display your Directory structure in the TREE command way.

When editing your program should provide the following facilities

- i. Insert an alphabet/number
- ii. Delete next alphabet/number
- iii. Backspace (delete previous alphabet/number)
- iv. Append at the end of line / file
- v. Multiline text
- vi. Movement of cursor, up, down, left, right
- vii. Redo (last 5 operations)
- viii. Undo (last 5 operations)

You are required to develop a text editor. Implement these features using data structures that you have already studied. (Hint: Doubly linked list, stack)

Example:

This assignment/Project is mainly contributed by Dr. Zeeshan Rana and in collaboration with Dr. Atif Mehdi. Thanks a lot for their help.

Open file test.txt. Type some text like, "This is a testing file". Press ENTER and a new line starts. Type some text like, "This is second line". Now your cursor is after "e" in the second line. Pressing backspace will delete "e" and the resultant text of second line will be, "This is second lin". Pressing left arrow key will move the cursor before "n". Pressing left arrow key two more times will move the cursor before "i" and then before "l". Now if "s" is typed, the result will be "This is second slin". Pressing delete two times will result in, "This is second sn". Now typing "u" will result, "This is second sun". Pressing CTRL-Z will undo the last operation and the result will be, "This is second sn". CTRL-Y will cause a redo of last operation, such as, "This is second sun". Only last 5 redo and 5 undo operations will be permitted.

Pressing up arrow will move the cursor to the start of the first row. Now typing "A" will result in, "A This is a testing file". Similarly, pressing down arrow will move the cursor to the start of second row. Pressing down arrow again will not have any effect as it is already the last row. Similar limit apply to up, left and right arrow keys.

7. Files enter the print queue at first come first serve basis; arrival time of each print request determines which request arrived first. A file is removed from the queue after its time to print expires. This virtual printing of a file cannot be interrupted once it has started (i.e. once a file has a turn, it cannot be removed from the queue until its time expires).
8. Two or more files with same name cannot exist in any of your virtual directory. However, the two files may happen to exist in same directory on your computer (For example, you might create all your files in C:\temp of your computer or your project folder). Suppose, you create two files with name a.txt in two different virtual directories; for example V:\X\Y\a.txt and V:\P\Q\a.txt; the files cannot be saved with the same name (i.e. a.txt) in c:\temp on your computer. To resolve this issue, you should name your files by adding a prefix of complete virtual directory path with your file name. V:\X\Y\a.txt will have its actual name as V_X_Y_a.txt, whereas name of V:\P\Q\a.txt will be V_P_Q_a.txt. So you need to use a mechanism for naming your files such that you can keep your files in same folder on your disk.
- ~~9. A file with higher priority comes on front of priority queue, given there are no files with higher priority at the front. Virtual printing of a file cannot be interrupted once it has started (i.e. once a file has a turn and its time has started, it cannot be removed from the queue until its time expires and its priority cannot be superseded). A file which has current turn to be printed is removed from the queue after its time to print expires.~~
10. CD can only work in case of directory name. In case a file name is provided to CD, it should print that the provided name is not a directory.

PART 2

Implement notepad vim (it should allow making files using the power of linked list). Where the entire document is a linked list of characters in each line at least. We will describe the details of this part during the class.

This assignment/Project is mainly contributed by Dr. Zeeshan Rana and in collaboration with Dr. Atif Mehdi. Thanks a lot for their help.