DSA - PROJECT

Folder Cleaner

→ You are to build a Folder Cleaner. It looks into old and redundant files (i.e., answer copy is present), old for more than N number of months, empty (i.e., no content), or not accessed for at least M number of times.

Team Id : Bitzbuddies
Project : P4

Team Member Detail :

Student Id	Name	contribution
202301178	Patel Prince	30%
202301147	Patel Jeinil	30%
202301174	Rathva Hardik	20%
202301063	Dabhi Vrajesh	20%

Pseudocode :

1. Define constants: N_MONTHS = 6, M_ACCESS_COUNT = 3
2. Define functions:
    - checkIfFileIsOld(filePath):
    - Get file information using stat()
    - Calculate months since last access
    - Print "File is old: filePath" if older than N_MONTHS
    - checkIfFileIsEmpty(filePath):
    - Open file using ifstream
    - If empty, print "File is empty: filePath" and delete file
    - deleteEmptyFiles(directory):
    - Open directory using opendir()
    - Iterate files:
    - If empty and regular file, delete it and print
    - deleteOldFiles(directory):
    - Open directory using opendir()
    - Iterate files:

- If old and regular file, delete it and print
- cleanFolder(folderPath):

- Define function compareFileSize that takes 'a' and 'b' as input Compare the size of 'a' with the size of 'b'. Return true if the size of 'a' is less than the size of 'b', otherwise return false

- Define function sortFilesBySize that takes 'directory' as input.Open the directory at 'directory'.Initialize an empty vector of FileData called 'files'.For each file in the directory. If the file is a regular file. Get the file's name and size. Add a new FileData object with the file's name and size to 'files'.Close the directory

- Sort 'files' in ascending order based on file size using compareFileSize.For each fileData in 'files'.Define function displayFileDates that takes 'directory' as input:.Open the directory at 'directory'.For each file in the directory.If the file is a regular file.Get the file's last modification time.Print "File: " concatenated with the file path

and ", Last Modified: " concatenated with the last modified time.Close the directory

- Delete empty files in folderPath
- Delete old files in folderPath

3. Define main():
   - Set folderPath to directory to clean
   - Call cleanFolder(folderPath)
4. In main():
   - Call cleanFolder() with folderPath

Time complexity :

Since Overall Time Complexity for :

→ cleanfolder function is O(n).
→checklfilesold Function is O(1).
→checklfFileisEmpty Function is O(1).
→checkFileAccessCount Function is O(1).

Here, we can see that the time complexity of the entire program is dominated by the cleanFolder Function which is O(n).

So, overall time complexity of the entire program will be O(n) where n is the number of files in the specified folder.

Space Complexity

→ the space complexity of the program is primarily determined by the variable and data structure used.

→ the main space-consuming structures are strings to store file paths and directory structures (DIR* and struct dirent*).

→Since these structures only hold references to file names and paths, and not the actual content of

the files, the space complexity is proportional to the number of files and directories in the specified folder ,which is O(n), where n is the number of files in the directory.

→Other variables used in the program have constant space complexity.

→ thus, the space complexity of the entire program is O(n), where n is number of the files in the folder.

Here we use sortfilesbysize function that sort the all files in folder by size in ascending order.here is an example of that code.

```
62 bytes        fullfile.txt
1308 bytes      sort by size.cpp
3911 bytes      try69999.cpp
4791 bytes      date.txt
5948 bytes      Bitzbuddies.cpp
5948 bytes      date.cpp
15831176 bytes  basic electrical engg v k mehta.pdf
19173623 bytes  lab .pdf
```

Here we use displayfilesdate function that show when the file is created in which folder path we give.

```
File: c:/Users/VICTUS/OneDrive/Desktop/study/MAYBE DSA/basic electrical engg v k mehta.pdf, Last Modified: Sat Sep 16 17:54:08 2023
File: c:/Users/VICTUS/OneDrive/Desktop/study/MAYBE DSA/Bitzbuddies.cpp, Last Modified: Sun Apr 07 22:17:24 2024
File: c:/Users/VICTUS/OneDrive/Desktop/study/MAYBE DSA/date.cpp, Last Modified: Sun Apr 07 21:54:06 2024
File: c:/Users/VICTUS/OneDrive/Desktop/study/MAYBE DSA/date.txt, Last Modified: Sun Apr 07 21:29:56 2024
File: c:/Users/VICTUS/OneDrive/Desktop/study/MAYBE DSA/fullfile.txt, Last Modified: Sun Apr 07 16:19:28 2024
File: c:/Users/VICTUS/OneDrive/Desktop/study/MAYBE DSA/lab .pdf, Last Modified: Wed Oct 18 10:06:04 2023
File: c:/Users/VICTUS/OneDrive/Desktop/study/MAYBE DSA/sort by size.cpp, Last Modified: Sun Apr 07 21:18:15 2024
File: c:/Users/VICTUS/OneDrive/Desktop/study/MAYBE DSA/try69999.cpp, Last Modified: Sun Apr 07 19:36:30 2024
```

Here we use DeleteemptyFiles function that delete the empty files.

```
PS C:\Users\VICTUS\OneDrive\Desktop\study\MAYBE DSA\output> & .\'Bitzbuddies.exe'
Deleted empty file: c:/Users/VICTUS/OneDrive/Desktop/study/MAYBE DSA/epmty1.txt
Deleted empty file: c:/Users/VICTUS/OneDrive/Desktop/study/MAYBE DSA/epmty2.txt
Deleted empty file: c:/Users/VICTUS/OneDrive/Desktop/study/MAYBE DSA/epmty3.txt
```

WHY WE USE SPECIFIC DATA STRUCTURE?

The provided code is used for cleaning up a folder by performing various operations on its files, such as deleting empty files, deleting old files, displaying file dates, and sorting files by size.

THE GIVEN CODE IS VERY RELIABLE TO USE AND UNDERSTAND AS FOR US.ALSO THE TIME COMPLEXITY OF WHOLE CODE IS O(N) AND FOR PARTICULAR FUNCTION THE TIME COMPLEXITY IS O(N).THE SPACE COMPLEXITY IS ALSO O(N). THEREFORE THE TIME AND SPACE COMPLEXITY IS LINEAR.