Project :- System Programming

Name :- Rahul Girish Bantode

Technology :- python

Objective :-

To Implement the directory traversal system using python (system is like a UNIX/LINUX or any operating system command line interpreter works)

Description :-

Implement the system like UNIX/LINUX commands line terminal which traverse the directory after passing the directory name to it and many other functions on the file systems. Implement this type of system using python technology or we can say that, to the system is file/directory manipulation and search the particular one.

The system supports following functionality :-

1. Application takes the directory name from user using command line prompt (the directory name/path should be absolute) if relative path is given then search application search for current working directory only. and after getting an input traverse through the directory and print all the files and subdirectory in it.

2. If user fails to give an input at the time of execution of application then system consider the current working directory and traverse through it.

3. The format of the files ,sub-directory in directory looks like as follow :-

   ++<main_directory>

         --<sub_directory 1>

         --<sub_directory 2>

         **<file_name 1>

         **<file_name 2>

   ++<sub_directory in main directory>

         **<files within directory>

4. The program should also supports the following command line options.

   a) –s or –S   If user entered this literal followed by directory path then he will displays the file name with its file size.

   b) –s or –S <file_size> If user entered this combination with directory path then he will displays the files names which size greater than input in size.

c) –f or –F <substring/extension> If user entered this combination followed by directory then he will displays the files names with specified extensions.
5. The program have one extension which accept the –s and –f i.e file size and file extension from user and display those files who satisfies above both criteria.

Commands and input guidelines:-

As a application need is to take all the input from the command line. So as a user you have take care of while giving an input through command line the way and format which I am going to discuss in this format / variation you have to give an input while executing the exe file.

Input while execution	Description
python <exe_name>	It's a basic file execution which doesn't take any input.
python <exe_name> <dir_path>	It's take the directory path along with executable to traverse over the directory.
python <exe> <dirpath> <-s>/<-S>	Its take directory path with –s/-S size literal to display files with their sizes.
python <exe> <dirpath> <-s>/<-S> 512	Its take directory path with –s/-S size literal combine with file size to find the files greater than its size.
python <exe> <dirpath> <-f> <substring>	This command to find the files with having inputed substring/extension.
python <exe> <dirpath> <-s> <-f> 512 ".py"	This command is the combination of file size and file extension and display those files which satisfies above conditions.

Application documentation and Testing:-

By using above commands the application will run very well, you just need to type those commands in proper format and directory file should be absolute.

To test the application I use one directory and runs the application for that directory along with documentation I will share you the output files too. And to test the application you can use your own test cases.

Sample input and output:-

when type the "dir" command on terminal the files/sub-directory look like:-

```
Command Prompt

D:\Assignments\15th_May>dir
 Volume in drive D has no label.
 Volume Serial Number is 0E74-2D29

 Directory of D:\Assignments\15th_May

27-05-2021  15:57    <DIR>          .
27-05-2021  15:57    <DIR>          ..
17-05-2021  10:15    <DIR>          Demo
23-05-2021  15:27    <DIR>          Duplicate_Files
17-05-2021  12:30    <DIR>          Hello
15-05-2021  20:40               511 script-1.py
15-05-2021  21:26               839 script-2.py
17-05-2021  17:28             1,404 script-3.py
16-05-2021  20:59             2,159 script-4.py
17-05-2021  10:32             2,522 script-5.py
23-05-2021  15:24             3,774 script-6.py
23-05-2021  15:05             5,599 script-7.py
               7 File(s)         16,808 bytes
               5 Dir(s)  88,667,193,344 bytes free

D:\Assignments\15th_May>
```

When our code executes the output looks like:-

```
Command Prompt
++ D:\Assignments\15th_May
        -- Demo
        -- Duplicate_Files
        -- Hello
        ** script-1.py
        ** script-2.py
        ** script-3.py
        ** script-4.py
        ** script-5.py
        ** script-6.py
        ** script-7.py
++ D:\Assignments\15th_May\Demo
        -- Hello
        -- Rahul
        ** file-1-Copy.txt
        ** file-2 - Copy (2).txt
++ D:\Assignments\15th_May\Demo\Hello
        -- Mahakal
        ** Hello-file-1.txt
++ D:\Assignments\15th_May\Demo\Hello\Mahakal
++ D:\Assignments\15th_May\Demo\Rahul
++ D:\Assignments\15th_May\Duplicate_Files
        ** duplicate_file_log-Mon_May_17_12-33-18_2021.txt
        ** duplicate_file_log-Mon_May_17_12-37-36_2021.txt
        ** duplicate_file_log-Mon_May_17_17-47-14_2021.txt
        ** duplicate_file_log-Mon_May_17_18-13-16_2021.txt
        ** duplicate_file_log-Mon_May_17_18-13-44_2021.txt
```

Output files with along with source code:-
1. output_when_none_arguments.txt
2. output_when_directory_path_provided.txt
3. output_find_size_of_file.txt
4. output_size_greater_than500bytes.txt
5. output_find_file_for_specified_extension.txt
6. output_combine_size_extensions.txt

There is two source file :-
1. directory_traversal.py :- which contains the source code and entry point of the application
2. module.py :- which contains all the function which are imported to directory_traversal.py main file

Note :- You need to keep those file in same folder, if you are keep it in another Folder then you need to change the import statement according to your folder name . I tried to make this application in modular way.

Conclusion :-

Implementation of system like the terminal which traverse over the directory/search over the particular file size or find the size of file greater than specified size. Or while doing all the thing through commands so in my application I take all the inputs from command line arguments.

By using this application user able to traverse and find the appropriate file with variation which metioned above.