DR.B. C. ROY ACADEMY OF PROFESSIONAL COURSES

(FORMERLY DR.B. C. ROY ENGINEERING COLLEGE)

BACHELOR OF COMPUTER APPLICATION (BCA)

- **Tittle** : File Organiser
- **Subtitle** : cli file organiser using Python3
- **Presented by** :- Arnab Kisku (32301222034)

  Sudarshan Mondal (32301222066)

  Aditya Sharma (32301222061)

  Shivangi Yadav (32301222017)

  Avik Samadder (32301222021)

Project Report: File Organizer

Project Overview

The File Organiser project aims to automate the organisation of files in a specified directory based on their file extensions. The program reads a mapping file that associates file extensions with destination directories and moves files accordingly. This project leverages Python's file handling and system interaction capabilities to achieve efficient file organisation.

Project Components

1. Source Directory Definition: The program takes a source directory as a command-line argument ('**sys.argv[1]**') to specify the location of the files to be organised.

2. File Extension Mapping: The program reads a text file ('**file_extensions.txt**') containing mappings of file extensions to destination directories. Each line in the file follows the format '`<extension>:<destination_directory>`'.

3. File Organisation Logic: The program iterates over the files in the source directory and checks their file extensions against the mapping. If a matching extension is found, the file is moved to the corresponding destination directory.

Implementation Details

- The program uses Python's '**os**' and '**shutil**' modules for file operations and directory creation.
- Error handling is implemented to manage scenarios such as missing mapping file or file read errors.
- Command-line arguments are utilised to specify the source directory, providing flexibility and ease of use.

Installation

1. Ensure you have Python 3.x installed on your system.
2. Download the '**file_organiser.py**' script to your local machine.
3. Download the '**file_extensions.txt**' file to your local machine.
4. Place both the files '**file_organiser.py**' & '**file_extensions.txt**' in same directory.
5. New file extensions : file type can be appended in the '**file_extensions.txt**' file.

Usage

To use the File Organiser, follow these steps:
1. Ensure Python 3.x is installed on your system.
2. Create a text file ('**file_extensions.txt**' if not already downloaded) with extension mappings.
3. Run the program using the command '`python3 fileOrganiserV3.py <source_directory>`'.

Example

Suppose `file_extensions.txt` contains mappings like:

```
.jpg:images
.pdf:documents
.py:source_code
```

To organise files in '**/path/to/source/directory**' :

```
$ Python3 file0rganiserV3.py '/path/to/source/directory'
```

The program will organise files in the specified source directory according to the mappings, creating destination directories if necessary.

Source Code

```
import os
import shutil
import sys

source_dir = sys.argv[1]

file_extensions = {}

try:
    with open('file_extensions.txt', mode='r') as file:
        for line in file:
            line = line.strip()
            if not line or ':' not in line:
                continue
            extension, category = line.split(':', 1)
            file_extensions[extension.strip()] = category.strip()
except FileNotFoundError:
    print("The file 'file_extensions.txt' was not found.")
    exit(1)
except Exception as e:
    print(f"An error occurred while reading the .txt file: {e}")
    exit(1)

for file in os.listdir(source_dir):

        file_name, file_extension = os.path.splitext(file)


        if file_extension in file_extensions:
                destination_dir = os.path.join(source_dir, file_extensions[file_extension])

        if not os.path.exists(destination_dir):
                os.makedirs(destination_dir)

        shutil.move(os.path.join(source_dir, file), os.path.join(destination_dir, file))

print('Files have been organized successfully!')
```

File Extension Mapping (file_extension.txt)

```
.jpg:images
.jpeg:images
.png:images
.gif:images
.bmp:images
.tiff:images
.svg:images
.pdf:documents
.doc:documents
.docx:documents
.xls:spreadsheets
.xlsx:spreadsheets
.ppt:presentation
.pptx:presentation
.txt:text
.csv:text
.html:web
.css:web
.js:web
.json:web
.xml:web
.zip:archives
.rar:archives
.7z:archives
.tar:archives
.gz:archives
.mp3:music
.wav:music
.flac:music
.aac:music
.ogg:music
.mp4:videos
.avi:videos
.mkv:videos
.mov:videos
.wmv:videos
.flv:videos
.mpeg:videos
.psd:graphics
.ai:graphics
.eps:graphics
.indd:graphics
.raw:graphics
.heic:images
.c:source_code
.cpp:source_code
.java:source_code
.py:source_code
.rb:source_code
.php:source_code
.swift:source_code
.go:source_code
.kt:source_code
.r:source_code
.m:source_code
.sql:database
.db:database
.sqlite:database
```

Conclusion

The File Organiser project simplifies file management tasks by automating the organisation process based on predefined rules. It enhances efficiency and reduces manual effort in maintaining a structured file system.