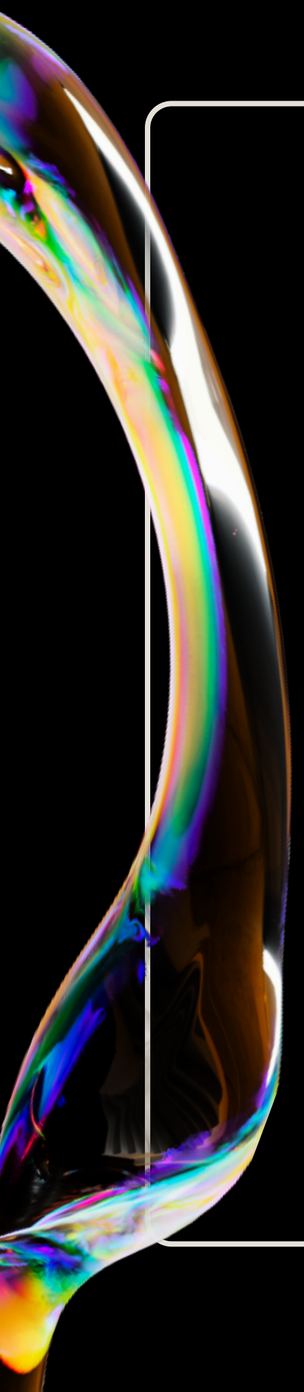
By Nyx Zhao

DIRECTORY SIZE CALCULATOR

zhaolili918@gmail.com

TABLE OF CONTENTS



INTRODUCTION

The Directory Size Calculator is a command-line interface (CLI) application that simulates a virtual file system with a hierarchical structure of directories and files.

It allows users to navigate the simulated file system and perform basic operations like changing directories, listing contents, and calculating directory sizes recursively.

1 APPROACH & DESIGN

FILE SYSTEM SET UP

The file system is represented in memory using two core classes:

- File: stores a file_name and a size attribute.
- Directory: stores a name, a list of File objects, a list of subdirectories, and a reference to its parent directory. It supports adding files and subdirectories.

.

CLI COMMANDS

I created three helper functions for the three core commands:

- cd – supports navigating into and out of subdirectories using relative paths (including .., ., and multi-level traversal like ../folder1/folder2)
- ls – lists both files and subdirectories
- size – recursively calculates the total size of the current directory including all nested subdirectories and files

MAIN LOOP LOGIC

The main.py file handles user input and delegates command logic to helper functions.

The CLI provides a prompt that reflects the current directory path, similar to a Unix shell.

2 KEY FILES

src/file_system.py

Defines the File and Directory classes.

Each file object has a file_name and size attribute.

Each directory has name, a list of files, a list of subdirectories, and a parent attribute. Methods include adding a file and adding a subdirectory to the directory.

src/main.py

Contains the main CLI loop and command logic.

Includes helper methods for handling basic versions of the commands cd, ls, and size.

src/example.py

Generates an example filesystem with nested directories and file structure for testing.

test/test_file_system.py

Unit testing for simulated CLI functionality

3 HOW TO TEST & RUN

1 2 3 4

Clone Repository

From Github, copy project URL and paste into terminal:

```
git clone <URL>
```

Go into project directory:

```
cd directory-size-calculator
```

Run Application

Press run for main.py in your IDE or run the command below into the terminal:

```
python3 src/main.py
```

try python if python3 does not work

use commands like cd, ls, and size to interact with the simulated file system

Run Unit Tests

Press run for test_file_system.py in your IDE or run the command below into the terminal:

```
python3 test/test_file_system.py
```

try python if python3 does not work

Customization

If you want to edit or expand the sample directory tree, open example.py and modify the build_example_filesystem() function.

If you want to edit or expand the unit tests, open test_file_system.py and modify or add your own functions to the unit test class.

4 TEST & SEED DATA

Seed Data

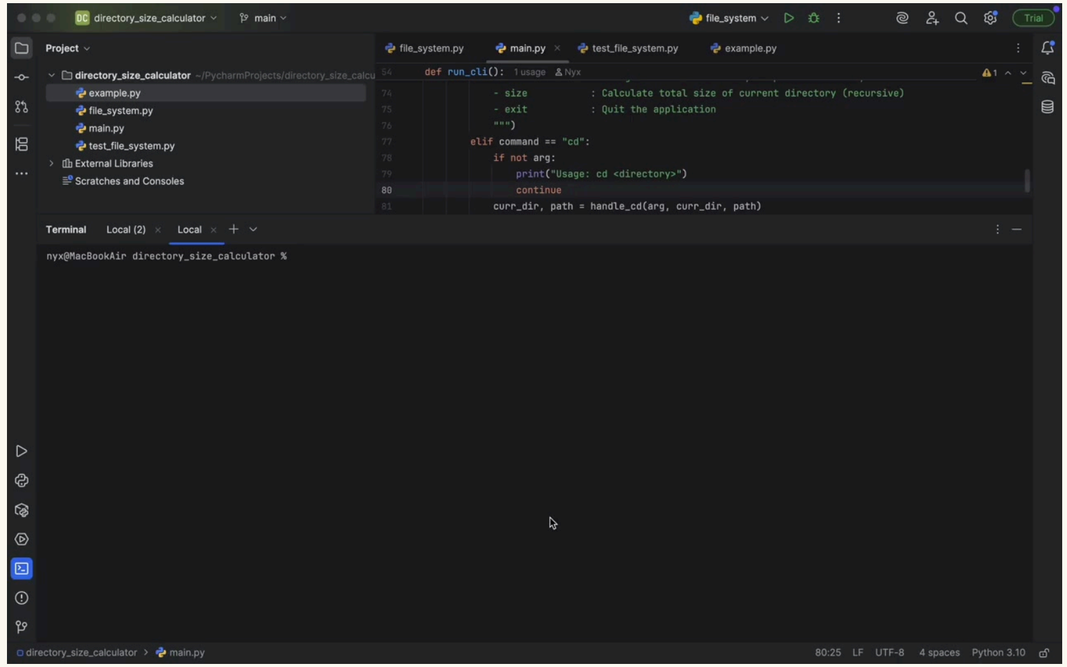
- An example file system is built in **example.py** with nested directories and sample files of varying sizes.

Test Data

- Unit tests in **test_file_system.py** verify:
  - Correct traversal behavior for the cd command (including invalid paths and nested navigation)
  - Accurate output of ls using captured stdout
  - Recursive correctness of the size function

- Additional edge cases like empty directories are also tested.

EXAMPLE APPLICATION RUN



By Nyx Zhao

THANK YOU

for your time and attention!

zhaolili918@gmail.com