

Applied Programming in Python
Assignment Lab 3

A simplified client – server solution for file management
Test Report

Group members:

1 – Mohammad Hossein Abdsharifi (19890907 - 5791)

2 – Ripan Kumar Dhar (19880727 - 8398)

Introduction

To verify the behavior and the outputs of all functions, and also make sure all the functions are working as intended, we provide a test file module to test our implementations in different aspects. In this module, we implemented 12 tests by using unittest library, however we did not test all functions and their outputs, but we tried to test major functions within 12 test function all under a class. In continue, we will briefly explain how the test module “file_test1.py” is conducted and how it works. During the coding, we tried to name the function according to their tasks to be clear as much as possible.

To be more clarify, the 12-test function and its explanations are follow:

- 1 – repr_test()
- 2 – login_register_test()
- 3 – change_directory_test()
- 4 – write_file_test()
- 5 – create_directory_test()
- 6 – list_file_test()
- 7 – read_noninput_test()
- 8 – write_noninput_test()
- 9 – create_write_test()
- 10 – read_write_test()
- 11 – readfile_empty_test()
- 12 – previous_directory_test()

Function Explanation:

1 – repr_test:

This function is designed to make sure the test module is set up properly and also the module is able to create new objects.

2 – login_register_test:

This function is implemented to check whether users are able to register usernames and passwords with privileges or not. In case the test function found any problem in registering function, it will show proper messages.

3 – change_directory_test:

The change directory function is implemented to check whether users can use change directory command with the correct way and also check whether the options for this command are working properly or not.

4 – write_file_test:

This function is used to check whether the user is able to create a text file for writing purposes or not. In case the function does not work as intended, a proper message will show.

5 – create_directory_test:

This function is conducted to check if the user can create directory in its working directory. After running the test file, a test folder will create to check the create directory in current working directory. To avoid mess up the root path, we removed any folder created by this function.

6 – list_file_test:

This test function is worked to check list function for listing all files and folders in current working directory. The function will check an empty file and listing an empty folder and returning a proper message.

7 – read_noninput_test:

The read noninput test is implemented to send read command without file name to check the read file options.

8 – write_noninput_test:

The write noninput test is implemented to send write command without file name to check the read file options.

9 – create_write_test:

This function is designed to test two functions, one for creating folder and one for write file to make sure those function is working as intended. Here also to avoid messing up the root path, we removed the test file we created for testing purposes.

10 – read_write_test:

This test function is designed to check whether the reading and writing functions are working properly. First, we created a text file to read and write on it by calling read and write file functions.

11 – readfile_empty_test:

In this function, we opened and read an empty file to make sure the read function will not crash in case the file is empty.

12 – previous_directory_test:

By this function, we put a test on change directory functions to make sure that users cannot leave their home directory by using two dots (..) many times.

Client – Server:

The server and client connection are only tested by asserts. But the conducted work is according to standard coding for socket programming with async, StreamReader and StreamWriter functions.