Computer Networks and Programming (ECE 5650)

Project 2

Fall 2020

Instructions:

- **Original Work**: this assignment must represent the original work of the team members.
    - You must not look at other solutions or show your solutions to anyone else.
    - You must not collaborate or discuss the assignment with other groups or individuals.
    - You must not get any portion of the code from Internet sources, other current or prior students, or other groups or individuals.
    - You must not ask or pay others to help you with the solution.

  Policy on Cheating, Fabrication, and Plagiarism: Cheating, fabrication, plagiarism, and helping others to commit these acts are all forms of academic dishonesty, and they are wrong. Therefore, avoid all appearance of improper behavior! Students who witness cheating should report the incident to the instructor as soon as possible. Academic misconduct will result in at least failing the course.

- **Late submissions:** No late submissions will be accepted.

- **Teamwork**: You must work with one other student and **must submit only one copy with both your names.**

- **Python version: your program(s) must be in Python 3.8.x or later**.

- **Socket programming: your program must use low-level socket programming for all communication/networking activities**. For these purposes, you must use only the socket library. For other purposes, you are free to use other libraries, such as sys, time, signal, os, etc.

- **Documentation:** Each required program must be well documented and well commented and must include the **authors and dates of creation and modification**.

- **Submission: please follow closely all submission instructions.**

- **Have questions?** Please contact the GTA/Grader. You can ask him by e-mail or send an e-mail to schedule a meeting on Microsoft Teams. No help will be given on the day the project is due.

Assessed Penalties:

Situation	**Penalty**
Late submission	Not accepted
Plagiarism or disallowed collaboration	At minimum negative grade
Not using low socket programming for all networking aspects	Not accepted
Not using Python 3.8.x or later	Not accepted
Failure to include the full report using the template and including the test procedure, thoughtful screenshots, and other requirements	Up to 40%
Failure to have a well commented and documented program(s), including the authors and dates of creation and modification	Up to 10%

Overview

This project will involve both research and programming. In this project, you will *extend* your Python Cloud File Processing programs in Project 2 by supporting multithreading and file compression, and then conducting thoughtful and detailed experiments, as explained next. **Your code must still support all operations required in Project 1.** The programs must be in **Python 3.8.x** or later and must use **your own low-level socket programming for all networking aspects**, as discussed in the class.

Assignment

- You will first need to read carefully and understand the technical details, specifications, and the examples of multithreading at https://docs.python.org/3.8/library/threading.html. The following tutorial will be helpful: http://www.tutorialspoint.com/python/python_multithreading.htm.

- (**50 Points**) Now, enhance your Cloud File Processing programs in Project 1 by
    - (40) Supporting multithreading. The server in Project 2 handles only one client at a time. Modify the program to implement a multithreaded server that is capable of serving multiple clients simultaneously. Using threading, first create a main thread in which your modified server listens for clients at a fixed port. When it receives a TCP connection request from a client, it will set up the TCP connection through

another port and services the client requests in a separate thread. There will be a separate TCP connection in a separate thread for each client-server pair.

- (10 Points) Supporting file compression. You must include a flag to enable compressing the file content before transmission to the other host. In that case, the other host must decompress the content before using it. Research for methods that can be used for compressing text files in Python and justify your choice in the report.

Your enhanced programs must be based on those in Project 1 and must still l support **ALL features** required by Project 1.

- (**10 Points**) Provide **thoughtful screenshots** to prove that your programs work as expected, with an adequate number of possible test cases. The **multithreading aspect** must be tested well. Be creative!

- (**40 Points**) Conduct thoughtful and detailed experiments to compare the performance of this multi-threaded server with that you developed in Project 2. Besides, compare the performance of enabling compression in the case of multithreading. You must decide on the proper performance metric to use, the type of experiments to be conducted, and how to conduct these experiments to quantitatively and adequately compare the performance of the two servers. In the report, you must specify your performance evaluation methodology, must include the results in both well-formatted and presented tables and graphs, and must briefly analyze the results. Use large files to easily compare the results.

Hints:

- When the server receives a request on the listening socket, it should create a connection socket (using the socket accept() method) and then pass this connection socket to a newly created thread. Thus, the accept() method should be called before the new thread is created, and then the connection socket must be passed as one of the parameters to the newly created thread. Ultimately, each client will be serviced by a different thread.

- To properly test multithreading and conduct the experiments, change the client code so that the requests are hardcoded (i.e. included as part of the code) rather than being requested from the user.

Submission Requirements

- Please follow carefully all submission requirements.

- **You must submit all the following files (a total of three files)**:
  - The client Python program in .py extension.
  - The server Python program in .py extension.

- **A professional report in pdf format, using the attached template and including the following sections:**
  a. **team member names,**
  b. **a copy of each source code,**
  c. **testing procedure used to verify the correctness of the program(s), including a description of inputs,**
  d. **screenshots and their explanations**,
  e. **completion status (self-critique)**, including answers to the following questions **for each required program**
     - Does your program meet all requirements? If not, explain the problem.
     - Does the program run correctly all the time? If not, explain the problem.
     - Did you adequately test the program? If not, specify.
     - Is the program well documented?
  f. **performance evaluation methodology, comparative results, and their analysis.**

- **File Naming:** the filenames must contain your last name(s): yourlastname1-yourlastname2-keyword.properExtension. The keyword describes the overall functionality of the program.
- The files must be submitted using **Canvas**:
  - Open this assignment on Canvas.
  - Select "Submit Assignment"
  - Select "File Upload".
  - Browse for the first file to upload and select it.
  - Choose "Add Another File" to upload the other files one at a time.
  - Hit the "Submit Assignment" button.

Good Luck!