CS4375: Theory of Operating Systems **Instructor: Dr. Deepak Tosh**
Due Date: 05/01/2025, Midnight

Assignment-6 (Multi-Threaded Socket-based File Server)

The goal of this part is to allow you exploring the applicability of sockets in creating a multi-threaded client-server application.

Problem Statement

Here, you are tasked to develop a multi-threaded server and a client program where the server can transfer a specific file with as many clients as possible in parallel. Assume that the server has a popular file called "***bio.txt***" and many clients are looking for downloading this file from a file server. Thus, the clients can make requests at any time by making a TCP connection and the server should reliably transfer ***bio.tx*t** to each client.

Things to consider

1. Use TCP sockets for both client and server.
2. The server should be able to handle multiple clients through threading.
3. Server reads the text file and can send **1000** bytes of data at a time. Thus, you can imagine, how many rounds of transmission to be done if the *bio.txt* is of 10MB size.
4. Client received the data bytes and appended into a file called "**received_bio_[clientid].txt**".
5. You are allowed to use either C or Python3 for developing this assignment.

After receiving the files, you should consider checking the differences between original bio.txt and received bio.txt, through a tool called "**diff**". There are also online tools to check the differences between two files. If the files do not match, then it tells that something in your program is not working correctly.

Note: You can try first with your own file that is small in size (may be < 1000bytes). Then, try the bio.txt file to transfer. You can take help from this link on understanding about multi-threaded client-server architectures: https://codezup.com/socket-server-with-multiple-clients-model-multithreading-python/. Also, feel free to explore more, there are tons of online resources on this.

To submit:

1. Two programs (server and client) with appropriate documentation and instruction to execute
2. Include any auxiliary files that you may have used.
3. A report on demonstrating your program's execution and correctness. Also mention any references that you used while developing your program.
4. Discussion on challenging aspects of this assignment.

Grading:

The assignment will be graded on following items:

1. Completeness and correctness on your responses, explanations, and observations.

2. Inclusion of appropriate evidence (in form of screenshots)
3. Clarity of report with description of each scenario
4. References (including links where you found some sample code).