<u>**Comp 2322 Computer Networking**</u>

Project: Multi-thread Web Server

Due time: 11:59pm, April 18, 2023, Tuesday

Total marks: 100 marks

Project objectives

This project aims to develop a socket program to implement a Web service using the HTTP protocol.

Design requirements

In this project, you are required to develop a multi-threaded Web server in Python that is capable of processing HTTP requests sent from browsers or some other client programs. This multi-threaded program will be able to handle multiple requests at the same time. Specifically, your Web server will

(i) create a connection socket when contacted by a client (browser);
(ii) receive the HTTP request from this connection;
(iii) parse the request to determine the specific file being requested;
(iv) get the requested file from the server's file system;
(v) create an HTTP response message consisting of the requested file preceded by header lines;
(vi) send the response over the TCP connection to the requesting client. If the client requests a file that is not present in your server, your server should return a "404 Not Found" error message.

Your task is to implement the server program, run your server program, and then test your server program by sending requests from the client programs running on different hosts. You may run the server on your own computer, using the IP address of 127.0.0.1. If you run your server on a host that already has a Web server running on it, then you should use a different port than port 80 for your Web server.

You can develop your code in two stages. In the first stage, you can simply implement the server program to receive the HTTP request messages and display the contents. After this is running properly, you can add the code to generate appropriate responses in the second stage. The Web server needs a log file to record statistics of the client requests. Each request corresponds to one line of record in the log. Write down client hostname/IP address, access time, requested file name and response type for each record. Your Web server also needs to handle some simple errors, such as web-page not found.

You can use either Python, Java or C/C++ languages for the project. When implementing the Web server, you are expected to use basic socket programming classes to build the Web server from scratch instead of using the HTTPServer class directly.

Submission requirements

Each student needs to submit a project package to Learn@PolyU, containing the following documents:

- A project report that contains
  - A cover page includes your name and student number;
  - A summary of your design and implementation of the server program;
  - A demonstration of executing your program and screen capturing of results of all functions;
  - A log file that records the historical information about the client requests and server responses.
- Complete source code
  - Your code should be commented appropriately.
- A README text file of how to compile and run your program

The due time of the project is 11:59pm, April 18, 2023, Tuesday, determined by Learn@PolyU. Do not challenge this time and submit your project package a little earlier. Late submission will cause the marks deducted 25% per day.

Assessment rubrics

The following rubrics will be used to evaluate your project quality and to determine your grade (100 marks + 10 bonus marks):

- Design and implement the Web server program to support the following functions (60 marks + 10 bonus marks)
  - Multi-threaded Web server (10 marks)
  - Proper request and response message exchanges (10 marks)
  - GET command for both text files and image files (10 marks)
  - HEAD command (10 marks)
  - Four types of response statuses ONLY, including 200 OK, 400 Bad Request, 404 File Not Found, 304 Not Modified (10 marks, marks will be deducted if your server program have returned more types of responses)
  - Handle Last-Modified and If-Modified-Since header fields (10 marks)
  - Handle Connection: Keep-Alive header field (10 bonus marks)
- Quality of your project's report (30 marks)
  - A good summary of your design and implementation of the server program (10 marks)
  - A full demonstration of executing your program and screen capturing of results of all functions (10 marks)
  - A complete log file (5 marks)
  - A clear README text file (5 marks)
- Quality of your project's source code (10 marks)
  - Complete source code for the project
  - Good naming and coding convention used in your source code
  - Compile the source code successfully
  - Execute the program without runtime errors

Comp 2322 Computer Networking
Project Marking Sheet

Student Information	**Name**		**Student ID**	
TA Information				
Check List				**Marks**
Design and implement the Web server program (60 marks+10 bonus marks)				
o Multi-threaded Web server (10 marks)				
o Proper request and response message exchanges (10 marks)				
o GET command for both text files and image files (10 marks)				
o HEAD command (10 marks)				
o Four types of response statuses ONLY (10 marks)				
o Handle Last-Modified and If-Modified-Since header fields (10 marks)				
o Handle Connection: Keep-Alive header field (10 bonus marks)				
Quality of your project’s report (30 marks)				
o A good summary of your design and implementation (10 marks)				
o A full demonstration of executing your program and screen capturing of results of all functions (10 marks)				
o A complete log file (5 marks)				
o A clear READMET text file (5 marks)				
Quality of your project’s source code (10 marks)				
o Complete source code for the project o Good naming and coding convention used in your source code o Compile the source code successfully o Execute the program without runtime errors				
Total				