Assigned: 2nd October, 2021
Due: 11th October, 2021

University of Missouri – St. Louis
Department of Mathematics and Computer Science

(Fall 2021: CMP SCI 4730 Computer Networks And Communications)

Lab Assignment 2B

Max Points: 40

1. **Socket programming exercise** [based on Assignment 1/Chapter 2 in the textbook]

This exercise involves development of a Web Server. This Web Server i) creates a connection when contacted by a client (browser), ii) receives HTTP requests from this connection; iii) parses the request to find the file being requested iv) get the file from the server file system and send the response over the TCP connection to the browser. If the requested file is not available in your server, your server should return, "404 file not found."

The skeleton code for the server is given below. Complete the code and run the server. Test it by sending requests from browsers.

To run the server

Put a test HTML file (e.g., HelloWorld.html) in the same directory that the server is in. Run the server program. Determine the IP address of the host that is running the server (e.g., 128.238.251.26). From another host, open a browser and provide the corresponding URL. For example: http://128.238.251.26:6789/HelloWorld.html 'HelloWorld.html' is the name of the file you placed in the server directory. Replace the port# 6789 with whatever port you have used in the server code (do not use reserved port numbers). The browser should then display the contents of HelloWorld.html [take a screenshot].

Try to get a file that is not present at the server. You should get a "404 Not Found" message [take a screenshot].

Submission

Submit both the screenshots and the completed code.

Skeleton code

```
from socket import *
import sys # to use system functions
# create a TCP server socket
serverSocket = socket(AF_INET, SOCK_STREAM)

#Fill in your code for binding and keeping your socket ready to listen
```

```
    while True:
        print("The server is ready to receive")

        connectionSocket, addr = #Fill your code for accepting client connection

        try:
            #receive message request
            message = #Fill your code here
            #Extract the path of the requested object from the message
            #The path is the second part of the HTTP header, identified by [1]
            filename = message.split()[1]
            #The extracted path of the HTTP requests include character '\', so read
            #the path from the second character
            f = open(filename[1:])
            #store the content of the requested file in a temporary buffer
            outputdata = f.read()

            #Send one HTTP header line into socket
            connection Socket.send("HTTP/1.1 200 OK\r\n\r\n".encode())
            #Send the content of the requested file to the client (connection socket)
            for i in range(0, len(outputdata)):
                connectionSocket.send(outputdata[i].encode())
            connectionSocket.send("\r\n".encode())

            #close the client connection socket
            connectionSocket.close()
        except IOError:
            #Send response message for file not found
            #Fill your code here for the '404 Not Found' message
            #Close client socket
            #Fill your code here
serverSocket.close()
sys.exit()
```

Note: You have a choice of writing your own HTTP client to test your server, instead of using a browser. Your client will connect to the server using a TCP connection, send an HTTP request to the server, and display the server response as an output. You can assume that the HTTP request sent is a GET method. The client should take command line arguments specifying the server IP address or host name, the port at which the server is listening, and the path at which the requested object is stored at the server.

The following is an input command format to run the client.

```
client.py server_host  server_port  filename
```