PROGRAMMING ASSIGNMENT 2

Summary:

The program created for Programming assignment 2 is explained step by step in this summary.

1. **In the first step, the required libraries were imported, in my code, the libraries are: socket , os , sys and _thread.**
2. **After that relevant codes were written to run socket programming successfully, eg: 'server_sock.accept'. Since TCP is used, 'SOCK_STREAM' is used instead of 'SOCK_DGRAM'.**
3. **Then, in this step codes were written to call a new thread, whenever the server receives a new request from it's clients (ie, browser).**
4. **After that a new user defined function called 'handler' is defined to handle requests from the client whenever a new thread is formed.**
5. **In the function 'handler', request from the client is received and processed. Here a While 'True' loop is used to maintain the server inside a loop.**
6. **In the 'handler' , the http request from the browser is split to obtain the path of the requested file or data. (.split() is used here)**
7. **Inside 'handler' , there is an if statement, which checks whether the client requests 'index.html', from the server. If that was the case, then 'index.html' is read by the server in bytes and sent to the client with proper http response structure (Content type text/html)**

8. In the else statement, the path of the requested file is transferred to a new user defined function called 'sender'.
9. Inside the 'sender' , the requested file is checked for it's presence inside 'www' folder, using the command 'os.path.isfile()'.
10. An if and else pair is used here. If the file is not found then an error message (status: 500) is sent to the browser inside the if statement.
11. If the requested file is found inside 'www' , then the control is transferred to the else statement. Here a nested if statement is used to send different header info , in the http response to browser corresponding to the extension of the requested file, the formats supported are .html, .css, .js, .png, .jpg, .gif and .txt.

- In this program the logic stated below is used to successfully send data, chunk by chunk to the browser (client).

```
Output_data = file.read(1024)
while(output_data):
  if(client_sock.send(output_data)):
    output_data = file.read(1024)
```