**Socket Programming Assignment 3 – Web Server**

CSC/CPE138 – Computer Network and Internet

Section 2: 9am – 10:15am

Professor Jun Dai

Vaukee Lee

**Goal:**

In this lab, we will be practicing more socket programming by developing a web server. These assignments are to help us review and apply your conceptual knowledge from the class. We will be using prior experience from the other socket programming assignments like the TCP client to server communication.

**Instructions:**

The first thing that we must do is to retrieve the skeleton code from the textbook’s companion website “[http://wps.pearsoned.com/ecs\_kurose\_compnetw\_6/](http://wps.pearsoned.com/ecs_kurose_compnetw_6/.).” For the web server, I will (1) create a connection socket when contacted by a client (browser); (2) receive the HTTP request from the connection; (3) parse the request to determine the specific file being requested; (4) get the requested file from the server's file system; (5) create an HTTP response message consisting of the requested file preceded by header lines; and (6) send the response over the TCP connection to the requesting browser. If a browser requests a file that is not present in your server, your server should return a "404 Not Found" error message. After this has all been completed, we will hope for the best and pray that our program runs. To make the server work, we will put in a .html file in the same folder.

The skeleton code looks like this:

#import socket module from socket import \*

serverSocket = socket(AF\_INET, SOCK\_STREAM)

#Prepare a sever socket

#Fill in start

#Fill in end

while True:

#Establish the connection

print 'Ready to serve...'

connectionSocket, addr = #Fill in start #Fill in end

try:

message = #Fill in start #Fill in end

filename = message.split()[1]

f = open(filename[1:])

outputdata = #Fill in start #Fill in end

#Send one HTTP header line into socket

#Fill in start

#Fill in end

#Send the content of the requested file to the client

for i in range(0, len(outputdata)):

connectionSocket.send(outputdata[i])

connectionSocket.close()

except IOError:

#Send response message for file not found

#Fill in start

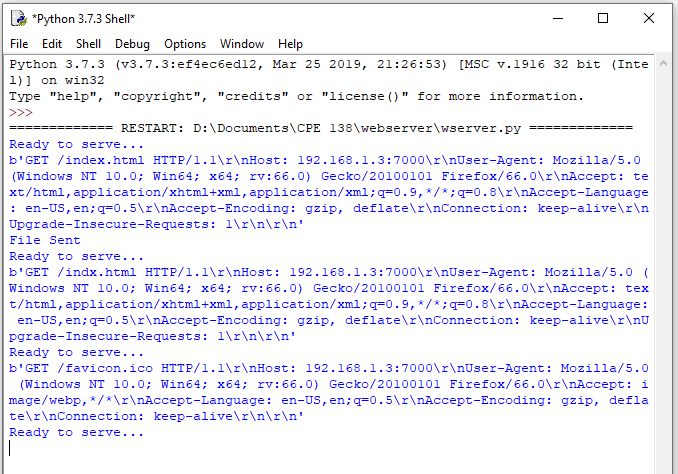
#Fill in end

#Close client socket

#Fill in start

#Fill in end

serverSocket.close()



This image is the Python shell that is outputting the data from the server. As you can see when the html file exists, the “File Sent” message appears but the when the file does not exist, the message header only show.



This image is my IP address along with the port address and the name of the .html file that I wanted to open. I also experimented with the 127.0.0.1 IP address.



This is how the website would look like my program worked properly.

**Conclusion:**

I ran into quite a bit of hiccups, along the way. The biggest one that I could not get to work was seeing the html file on the browser. The next one was this constant error that kept appearing. This error came out like this “TypeError: a bytes-like object is required, not 'str'.” I found out that with the newer version of Python (3.5), this error wouldn’t have occurred if I had Python 2.7 installed. To resolve this issue, I have to use the encode function or the letter “b” in front of the text. You can see this in my code. Overall this lab was straight-forward since most of the stuff we used were from the previous socket programming assignments. I tried to trouble shoot why the html file was not showing using the Wireshark program, but I didn’t see anything. Wireshark told me that it was getting the index.html file from the right location too.