Anthony Chavez

Professor Sun

Socket Programming Lab 2

HTTP Web Server Source Code

```
from socket import *
from time import sleep

#Prepare the sever socket
serverPort = 12003
serverSocket = socket(AF_INET, SOCK_STREAM)
serverSocket.bind((", serverPort))
serverSocket.listen(1)

#Print startup sequence
print ("Starting up server", end = " ", flush = True)
sleep(.5)
print (".", end = " ", flush = True)
sleep(.5)
print (".", end = " ", flush = True)
sleep(.5)
print (".", end = " ", flush = True)
sleep(.5)
print (". The server is ready to receive!")

while True:
        #Establish the connection
        connectionSocket, addr = serverSocket.accept()
        try:
                message = connectionSocket.recv(1024)
                print(message)

                #Parse message for requested file name
                fileName = message.split()[1]
                print(fileName)

                #Open and read file
                f = open(fileName[1:])
                outputdata = f.read()

                #Send one HTTP header line into socket
                connectionSocket.send(b'HTTP/1.1 200 OK\r\n\r\n')

                #Send the content of the requested file to the client
                for i in range(0, len(outputdata)):
                        connectionSocket.send((outputdata[i]).encode())
                #Close client socket
                connectionSocket.close()

                #Close client socket
                connectionSocket.close()
```

```
    except IOError:
        #Send response message for file not found
        connectionSocket.send(b'HTTP/1.1 404 Not Found\r\n\r\n')
        connectionSocket.send(b'404 Not Found')
        #Close client socket
        connectionSocket.close()
serverSocket.close()
```

HTTP Client Source Code

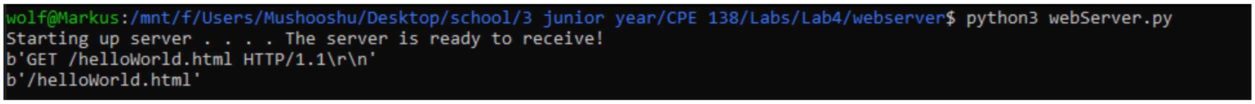
```
from socket import *
from time import sleep
import sys

#Prepare the client socket
serverName = '10.0.0.236'
serverPort = 12003
clientSocket = socket(AF_INET, SOCK_STREAM)

#Try 3-Way Handshake
try:
    clientSocket.connect((serverName, serverPort))
except:
    print("Server is currently busy or offline. Please try again later.")
    clientSocket.close()
    sys.exit()
print("Connection Established.")

#Ask User for a file
fileName = input("Please enter the file you wish to request: ")

#Send HTTP request
request = "GET /" + fileName + " HTTP/1.1\r\n"
clientSocket.send(request.encode())
print("HTTP request was sent.")

#Receiving HTTP response
print("Server Response:\r\n")
data = ""
while True:
    newData = clientSocket.recv(1024).decode()
    data += newData
    if(len(newData) == 0):
        break
print(data)
```

```
#Close client socket and end program
print("Closing socket ", end = " ", flush = True)
sleep(.5)
print (". ", end = " ", flush = True)
sleep(.5)
print (". ", end = " ", flush = True)
sleep(.5)
print (". ", end = " ", flush = True)
sleep(.5)
print (". Goodbye")
clientSocket.close()
```

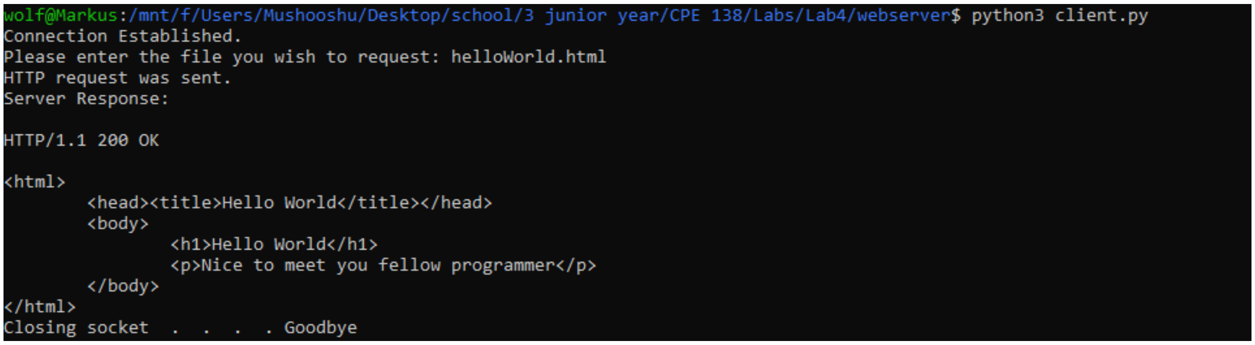
helloWorld.html Source Code

```
<html>
        <head><title>Hello World</title></head>
        <body>
                <h1>Hello World</h1>
                <p>Nice to meet you fellow programmer</p>
        </body>
</html>
```

Execution of source code in terminal

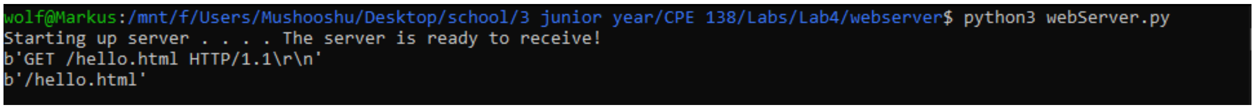


```
wolf@Markus:/mnt/f/Users/Mushooshu/Desktop/school/3 junior year/CPE 138/Labs/Lab4/webserver$ python3 webServer.py
Starting up server . . . . The server is ready to receive!
b'GET /helloWorld.html HTTP/1.1\r\n'
b'/helloWorld.html'
```

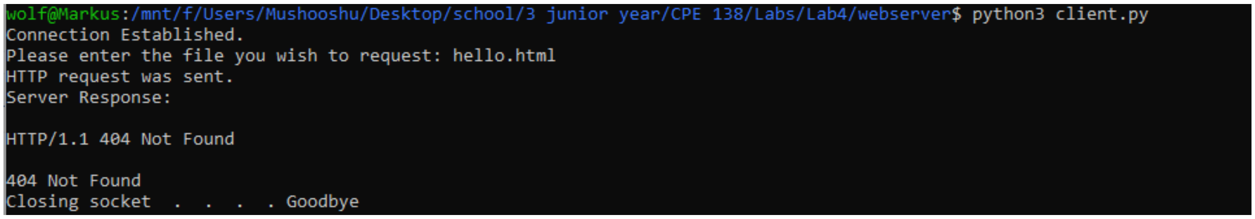


```
wolf@Markus:/mnt/f/Users/Mushooshu/Desktop/school/3 junior year/CPE 138/Labs/Lab4/webserver$ python3 client.py
Connection Established.
Please enter the file you wish to request: helloWorld.html
HTTP request was sent.
Server Response:

HTTP/1.1 200 OK

<html>
        <head><title>Hello World</title></head>
        <body>
                <h1>Hello World</h1>
                <p>Nice to meet you fellow programmer</p>
        </body>
</html>
Closing socket  .  .  .  . Goodbye
```

In the first snippet, we can see the output of the server code. The first line tells the User the server is online and ready to receive requests. The second line is the request from the client (browser). The third line shows filename being requested. In the second snippet, we can see the server response. As expected, a 200 OK status code was received along with the data in the html file since the file exists at the webserver.



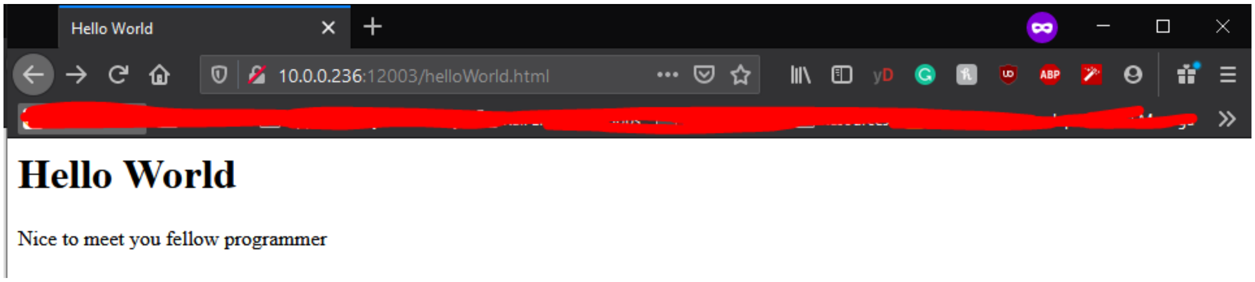
```
wolf@Markus:/mnt/f/Users/Mushooshu/Desktop/school/3 junior year/CPE 138/Labs/Lab4/webserver$ python3 webServer.py
Starting up server . . . . The server is ready to receive!
b'GET /hello.html HTTP/1.1\r\n'
b'/hello.html'
```



```
wolf@Markus:/mnt/f/Users/Mushooshu/Desktop/school/3 junior year/CPE 138/Labs/Lab4/webserver$ python3 client.py
Connection Established.
Please enter the file you wish to request: hello.html
HTTP request was sent.
Server Response:

HTTP/1.1 404 Not Found

404 Not Found
Closing socket  .  .  .  . Goodbye
```

In the first snippet, we can see the output of the server code. The first line tells the User the server is online and ready to receive requests. The second line is the request from the client (browser). The third line shows filename being requested. In the second snippet, we can see the server response. As expected, a 404 Not Found status code was received since the file does not exist at the webserver.

Displaying file in the web browser



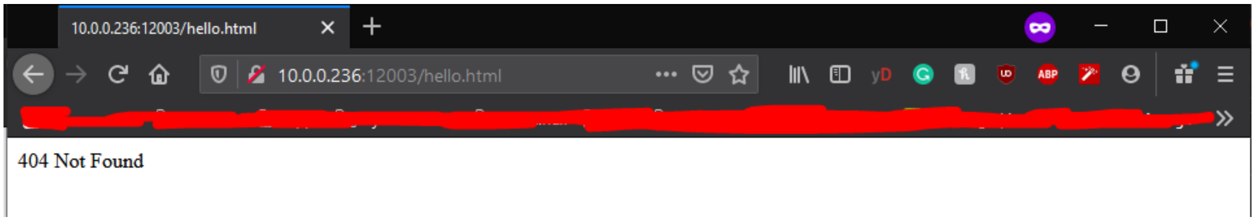
[redacted]

Hello World

Nice to meet you fellow programmer

Please note that the webserver.py program must be running to see html file in the web browser. Using the URL, http://10.0.0.236:12003/helloWorld.html, we can see the browser successfully receives the html file and displays the contents.

Displaying "404 Not Found" message in web browser



[redacted]

404 Not Found

Please note that the webserver.py program must be running to see the html file in the web browser. Using the URL, http://10.0.0.236:12003/hello.html, we can see the browser is unsuccessful in receiving the html file and displays a "404 Not Found" message.