CSE 476/ CSE 575
Fall 2020

Term Project Assignment Report

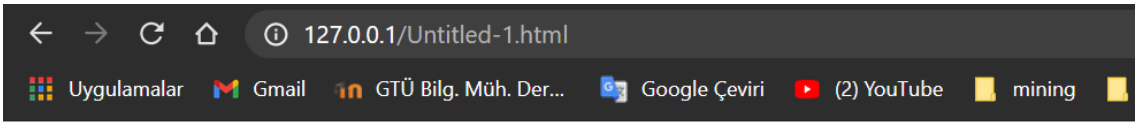
161044005

Feyza Nur Akyol

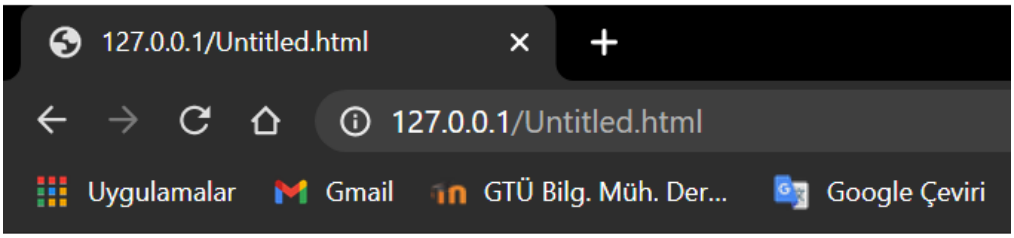
161044005
FEYZA NUR AKYOL

Assignment 1: Web Server

My program creates a socket, then creates a connection with a browser. Receives it and then parse as then gets a html content. But my program doesn't understand the content as a html, it sees as a txt file. I figure out that I should add some extras into header part of the html. But I don't have time to fix the problem.




```
Content-Type: text/html/r/n<!DOCTYPE HTML>
<html lang="en">
<head>
<META charset="UTF-8">
<META name="viewport" content="width=device-width, initial-scale=1.0">
<title>Sample Web Page</title>
</head>
<body>
(Contents go here)

</body>
</html>
```




```
404 Not Found
```

```
Ready to serve...

404 Not Found


Ready to serve...
```

➔ Here prints result when the file does not exist.

CODE:

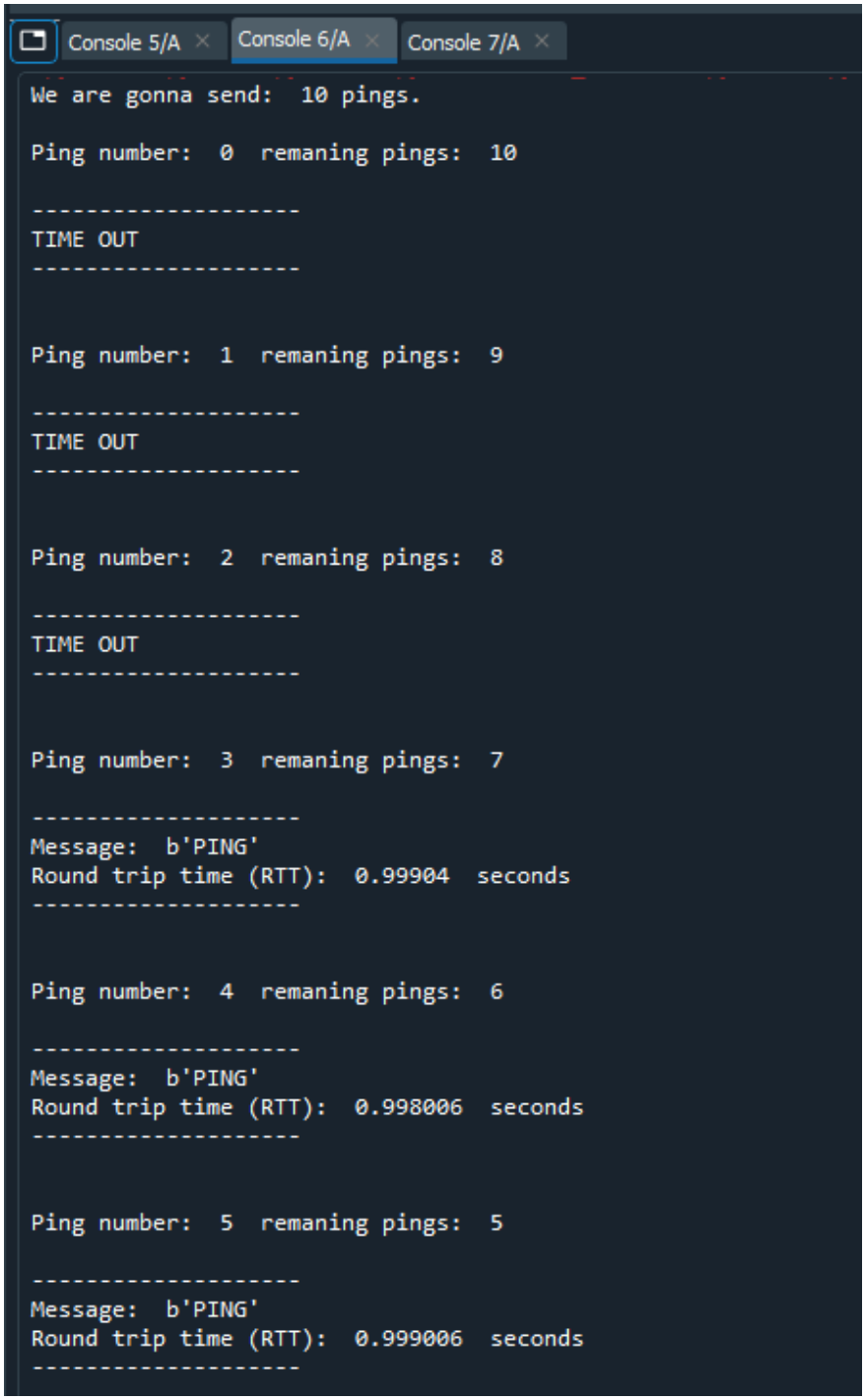
```
from socket import *

serverSocket = socket(AF_INET, SOCK_STREAM)
#Prepare a sever socket
serverSocket.bind(('',80))
serverSocket.listen(10)
while True:
    #Establish the connection
    print ('Ready to serve...')
    connectionSocket, addr = serverSocket.accept()
    try:
        message = connectionSocket.recv(1024).decode('utf-8')
        filename = message.split()[1]
        f = open(filename[1:])
        outputdata = f.read()
        print(outputdata)

        #Send one HTTP header line into socket
        connectionSocket.sendall(b'HTTP/1.1 200 OK\r\n\r\n')
        connectionSocket.sendall(b'Content-Type: text/html/r/n')

        #Send the content of the requested file to the client
        for i in range(0, len(outputdata)):
            connectionSocket.sendall(outputdata[i].encode('utf-8'))
        connectionSocket.close()

    except IOError:
        #Send response message for file not found
        connectionSocket.sendall(b'404 Not Found')
        print("\n404 Not Found\n\n")

        #Close client socket
        connectionSocket.close()

serverSocket.close()
```

161044005
FEYZA NUR AKYOL

Assignment Part 2:

My ping program is to send 10 ping messages to the target server over UDP. For each message, my client is calculate and print the RTT when the corresponding pong message is returned. My client wait up 1 second for a reply, if there is no reply prints error message.

I tested my code in 127.0.0.1 address and it worked properly. Here are results;



```
Ping number:  6  remaning pings:  4

--------------------
Message:  b'PING'
Round trip time (RTT):  0.99899  seconds
--------------------


Ping number:  7  remaning pings:  3

--------------------
TIME OUT
--------------------


Ping number:  8  remaning pings:  2

--------------------
TIME OUT
--------------------


Ping number:  9  remaning pings:  1

--------------------
Message:  b'PING'
Round trip time (RTT):  0.999005  seconds
--------------------


Ping number:  10  remaning pings:  0

--------------------
TIME OUT
--------------------
```

Server Code:

```
"""
Created on Mon Nov 30 13:29:24 2020

@author: feyzanur.akyol2016
"""
import random
from socket import *

serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind(('',12000))

while True:
    rand = random.randint(0,10)
    message, address = serverSocket.recvfrom(1024)
    message = message.upper()
    if rand < 4:
        continue
    serverSocket.sendto(message, address)
```

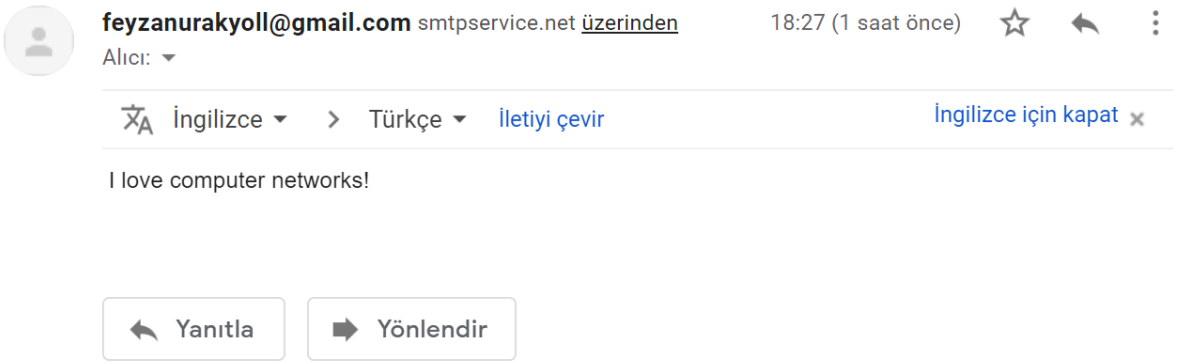
Client Code:

```
"""
Created on Mon Nov 30 13:37:26 2020

@author: feyzanur.akyol2016
"""

from datetime import datetime
from socket import *
from time import time

def main():
    servername = 'localhost'
    serverport = 12000
    clientSocket = socket(AF_INET,SOCK_DGRAM)
    message = 'Ping'
    counter = 10
    print ('We are gonna send: ', counter, 'pings.\n')
    for i in range(counter+1):

        print('Ping number: ', i, ' remaning pings: ',(counter-i))

        timeOne = datetime.now()
        clientSocket.sendto(message.encode('utf-8'), (servername,serverport))
        clientSocket.settimeout(1)

        try:
            modifiedMessage, serverAddress = clientSocket.recvfrom(1024)
            timeTwo = datetime.now()
            timeRemain = timeOne - timeTwo
            print ("\n--------------------\nMessage: ",modifiedMessage)
            print('Round trip time (RTT): ', (timeRemain.microseconds/1000000)
 , ' seconds\n--------------------\n\n')

        except timeout:
            print('\n--------------------\nTIME OUT\n--------------------
\n\n')

    if i == 10:
        clientSocket.close()

pass
if __name__ == '__main__':
    main()
```

Assignment Part 3:

In this part, I used ‘SMTP2GO’ mail server. I created a socket and started a connection. I used ‘ehlo’ instead ‘helo’ because I got "Must issue a STARTTLS command first" error. My authantication is accepted too, then i sent all need parts like; mail from, rcpt to etc. Then I quit and close the socket.

```
Recived Message:  b'220 mail.smtp2go.com ESMTP Exim 4.92-S2G Mon, 04 Jan 2021
17:27:45 +0000\r\n'

HELO command server response:  b'250-mail.smtp2go.com Hello Alice [78.190.139.86]
\r\n250-SIZE 52428800\r\n250-8BITMIME\r\n250-DSN\r\n250-PIPELINING\r\n250-AUTH
CRAM-MD5 PLAIN LOGIN\r\n250-CHUNKING\r\n250-STARTTLS\r\n250-PRDR\r\n250 HELP\r\n'

AUTHANTICATION: 235 Authentication succeeded


MAIL FROM :250 OK


RCPT TO: 250 Accepted <fnuakyol@gmail.com>


DATA: 354 Enter message, ending with "." on a line by itself


ENDS : 250 OK id=1kwTdy-Duub28-CB


QUIT 221 mail.smtp2go.com closing connection
```



161044005
FEYZA NUR AKYOL

Code:

```
from socket import *
import base64
import ssl

msg = "\r\n I love computer networks! \r\n\r\n"
endmsg = "\r\n.\r\n"

# Choose a mail server (e.g. Google mail server) and call it mailserver
mailserver = ("mail.smtp2go.com", 2525)

# Create socket called clientSocket and establish a TCP connection with mailse
rver
clientSocket = socket(AF_INET,SOCK_STREAM)
clientSocket.connect(mailserver)

recv = clientSocket.recv(1024)
print ("Recived Message: ", recv)
if recv[:3] != b'220':
    print ("220 reply not received from server.\n")

# Send HELO command and print server response.
#I used EHLO instead HELO, because I got "Must issue a STARTTLS command first"
#error
helo = 'EHLO Alice\r\n'
clientSocket.send(helo.encode())
recv1 = clientSocket.recv(1024)
print ("\nHELO command server response: ",recv1)
if recv1[0:3] != b'250':
    print ("250 reply not received from server.\n")


#Username and password information (mail and password is not correct)
username = "example_smtp2go@example.com"
password = "example_Password"
base64_str = ("\x00"+username +"\x00"+password).encode()
base64_str =  base64.b64encode(base64_str) #bu ne yapıyo bi baksana
authentMessage = "AUTH PLAIN ".encode() + base64_str + "\r\n".encode()
clientSocket.send(authentMessage)
receiveAuth = clientSocket.recv(1024)
print("\nAUTHANTICATION: " + receiveAuth.decode())
if receiveAuth[0:3] != b'235':
    print ("235 reply not received from server.\n")

# Send MAIL FROM command and print server response.
mailFrom = "MAIL FROM: <feyzanurakyoll@gmail.com> \r\n"
```

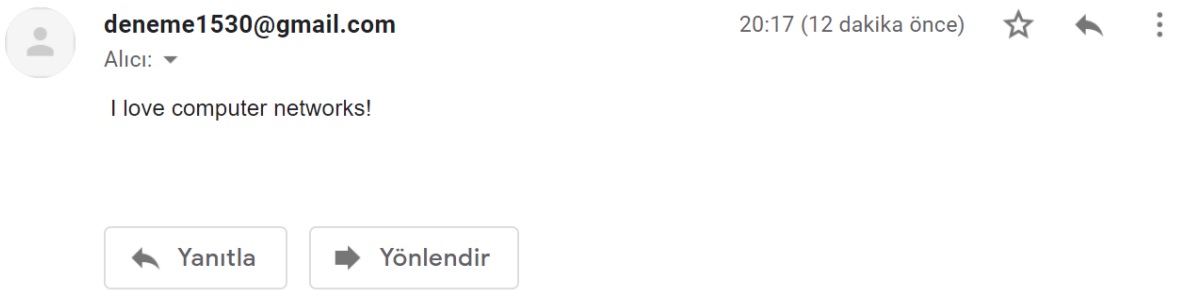
```
clientSocket.send(mailFrom.encode())
recvMail = clientSocket.recv(1024)
print ("\nMAIL FROM :" + recvMail.decode())
if recvMail[0:3] != b'250':
    print ("250 reply not received from server.\n")

# Send RCPT TO command and print server response.
rcptTo = "RCPT TO: <fnuakyol@gmail.com> \r\n"
clientSocket.send(rcptTo.encode())
recvRcp = clientSocket.recv(1024)
print("\nRCPT TO: " + recvRcp.decode())
if recvRcp[0:3] != b'250':
    print ("250 reply not received from server.\n")

# Send DATA command and print server response.
data = "DATA \r\n"
clientSocket.send(data.encode())
recvData = clientSocket.recv(1024)
print("\nDATA: " + recvData.decode())
if recvData[0:3] != b'354':
    print ("250 reply not received from server.\n")

# Send message data.
givenMsg = "Enter your message: "
clientSocket.send((givenMsg + msg).encode())
#I didn't receive a request message, cuz data is not ended yet.


# Message ends with a single period.
clientSocket.send(endmsg.encode())
recv_message1 = clientSocket.recv(1024)
print ("\nENDS : " + recv_message1.decode())
if recv_message1[0:3] != b'250':
    print ("250 reply not received from server.\n")

# Send QUIT command and get server response.
clientSocket.send("QUIT \r\n".encode())
message=clientSocket.recv(1024)
print ("\nQUIT " + message.decode())
if message[0:3] != b'221':
    print ("221 reply not received from server.\n")
clientSocket.close()
```

Part 3 : Bonus

I used *gmail* mail server instead of *smtp2go*. Gmail needs to connection more secure. So I used 'STARTTLS' and also ssl.wrapsocket(). Other parts were same.

Here is my results:

```
Recived Message:  b'220 smtp.gmail.com ESMTP w20sm44598162edi.12 - gsmtp\r\n'

HELO command server response:  b'250 smtp.gmail.com at your service\r\n'
STARTTLS Response: 220 2.0.0 Ready to start TLS


AUTHANTICATION: 235 2.7.0 Accepted


MAIL FROM :250 2.1.0 OK w20sm44598162edi.12 - gsmtp


RCPT TO: 250 2.1.5 OK w20sm44598162edi.12 - gsmtp


DATA: 354  Go ahead w20sm44598162edi.12 - gsmtp


ENDS : 250 2.0.0 OK  1609780787 w20sm44598162edi.12 - gsmtp


QUIT 221 2.0.0 closing connection w20sm44598162edi.12 - gsmtp
```



Code:

```
"""
Created on Mon Jan  4 20:17:44 2021

@author: Feyza
"""

from socket import *
import base64
import ssl

msg = "\r\n I love computer networks! \r\n\r\n"
endmsg = "\r\n.\r\n"

# Choose a mail server (e.g. Google mail server) and call it mailserver
mailserver = ("smtp.gmail.com", 587)


# Create socket called clientSocket and establish a TCP connection with mailse
rver
clientSocket = socket(AF_INET,SOCK_STREAM)
clientSocket.connect(mailserver)

recv = clientSocket.recv(1024)
print ("Recived Message: ", recv)
if recv[:3] != b'220':
    print ("220 reply not received from server.\n")


# Send HELO command and print server response.
helo = 'HELO Alice\r\n'
clientSocket.send(helo.encode())
recv1 = clientSocket.recv(1024)
print ("\nHELO command server response: ",recv1)
if recv1[0:3] != b'250':
    print ("250 reply not received from server.\n")

#Send STARTTLS  command
clientSocket.send(("STARTTLS\r\n".encode()))
recvTls = clientSocket.recv(1024)
print("STARTTLS Response: " + recvTls.decode())
clientSocket = ssl.wrap_socket(clientSocket)


#Username and password information
username = "deneme1530@gmail.com"
```

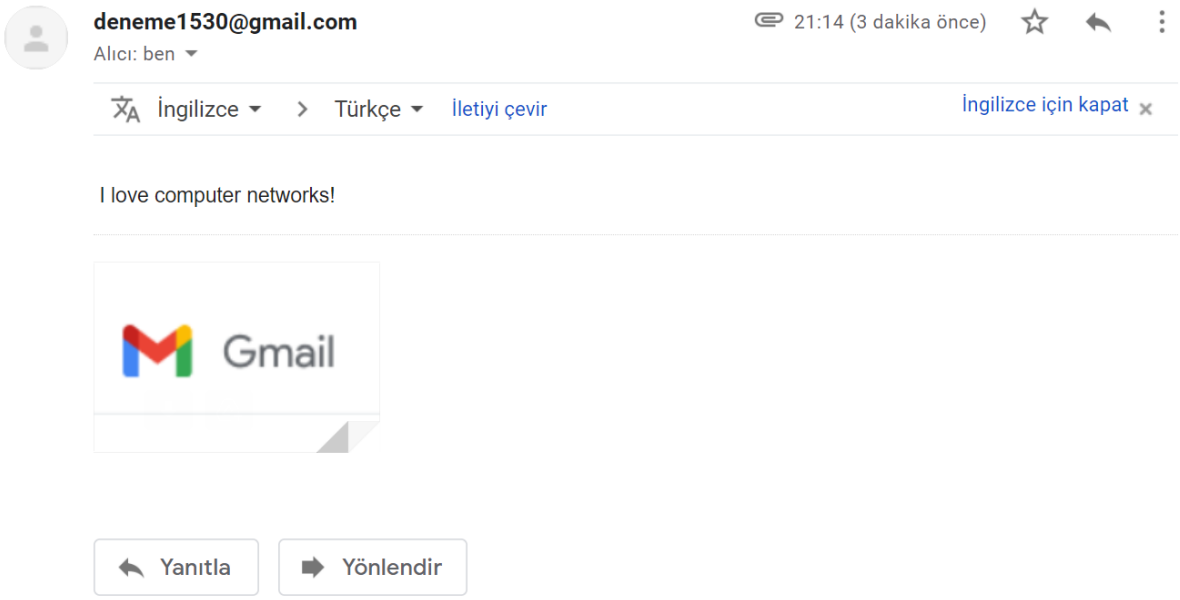
```
password = "EK123456"
base64_str = ("\x00"+username +"\x00"+password).encode()
base64_str =  base64.b64encode(base64_str) #bu ne yapıyo bi baksana
authentMessage = "AUTH PLAIN ".encode() + base64_str + "\r\n".encode()
clientSocket.send(authentMessage)
receiveAuth = clientSocket.recv(1024)
print("\nAUTHANTICATION: " + receiveAuth.decode())
if receiveAuth[0:3] != b'235':
    print ("235 reply not received from server.\n")

# Send MAIL FROM command and print server response.
mailFrom = "MAIL FROM: <feyzanurakyoll@gmail.com> \r\n"
clientSocket.send(mailFrom.encode())
recvMail = clientSocket.recv(1024)
print ("\nMAIL FROM :" + recvMail.decode())
if recvMail[0:3] != b'250':
    print ("250 reply not received from server.\n")

# Send RCPT TO command and print server response.
rcptTo = "RCPT TO: <fnuakyol@gmail.com> \r\n"
clientSocket.send(rcptTo.encode())
recvRcp = clientSocket.recv(1024)
print("\nRCPT TO: " + recvRcp.decode())
if recvRcp[0:3] != b'250':
    print ("250 reply not received from server.\n")

# Send DATA command and print server response.
data = "DATA \r\n"
clientSocket.send(data.encode())
recvData = clientSocket.recv(1024)
print("\nDATA: " + recvData.decode())
if recvData[0:3] != b'354':
    print ("250 reply not received from server.\n")

# Send message data.
givenMsg = "Enter your message: "
clientSocket.send((givenMsg + msg).encode())
#I didn't receive a request message, cuz
#data is not ended yet.


# Message ends with a single period.
clientSocket.send(endmsg.encode())
recv_message1 = clientSocket.recv(1024)
print ("\nENDS : " + recv_message1.decode())
if recv_message1[0:3] != b'250':
    print ("250 reply not received from server.\n")

# Send QUIT command and get server response.
```

```
clientSocket.send("QUIT \r\n".encode())
message=clientSocket.recv(1024)
print ("\nQUIT " + message.decode())
if message[0:3] != b'221':
    print ("221 reply not received from server.\n")
clientSocket.close()
```

Part 3: Bonus 2

I sent my message using as multipart messages. In this type our message consists of multiple messages. Also we can define their content type. To do that I used MIMEMultipart class.

```
Mail is sended!
```



Code:

```
from socket import *
import base64
import ssl
import smtplib
from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText
from email.mime.base import MIMEBase
from email import encoders

# instance of MIMEMultipart
message = MIMEMultipart()
message['From'] = "deneme1530@gmail.com"
message['To'] = "fnuakyol@gmail.com"
message['Subject'] = "Image Sending"
body = "\r\n I love computer networks! "

# Add Image
message.attach(MIMEText(body, 'plain'))

# open the file to be sent
filename = "Image.png"
attachment = open("Image.png", "rb")
image = MIMEBase('application', 'octet-stream')
image.set_payload((attachment).read())

# encode
encoders.encode_base64(image)

#add filename and positional argument in header
image.add_header('Content-
Disposition', "attachment; filename= %s" % filename)

# add image to the message
message.attach(image)

# then start
server = smtplib.SMTP('smtp.gmail.com', 587)
server.starttls()
server.login("deneme1530@gmail.com", "EK123456")
text = message.as_string()
server.sendmail("deneme1530@gmail.com", "fnuakyol@gmail.com", text)
print("\nMail is sended!")
server.quit()
```