GTU Department of Computer Engineering
CSE 476/575 Mobile Communication Networks
Fall 2021 - Term Project Assignments Report

Akif KARTAL
171044098

Assignment 1: Web Server

Prepare a sever socket

```
# Fill in start
serverPort = 5000
serverSocket.bind(('', serverPort))
serverSocket.listen(1)
print('server running on 127.0.0.1:', serverPort)
# Fill in end
```

*As you can see, port is 5000.

Start to listen connection

```
print('Ready to serve...')
connectionSocket, addr = serverSocket.accept()  # Fill in start             #Fill in end
```

Read coming data

```
message = connectionSocket.recv(1024)  # Fill in start
filename = message.split()[1]
f = open(filename[1:])
outputdata = f.read()  # Fill in start       #Fill in end
# Send one HTTP header line into socket
```

Send Success messages

```
# Fill in start
connectionSocket.send('\nHTTP/1.1 200 OK\n\n'.encode())
# Fill in end
# Send the content of the requested file to the client
for i in range(0, len(outputdata)):
    connectionSocket.send(outputdata[i].encode())
connectionSocket.close()
```

Send Error messages

```
# Send response message for file not found
# Fill in start
connectionSocket.send("\nHTTP/1.1 404 Not Found\n\n".encode())
# Fill in end
# Close client socket
# Fill in start
connectionSocket.close()
# Fill in end
```

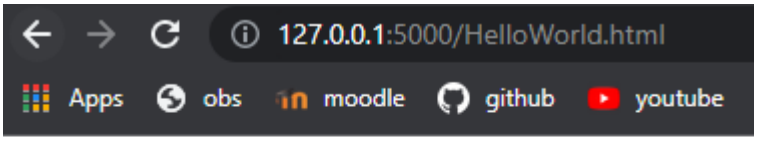
Testing

In order to test, I create a simple HelloWorld.html file.

Start

```
server running on 127.0.0.1: 5000
Ready to serve...
```

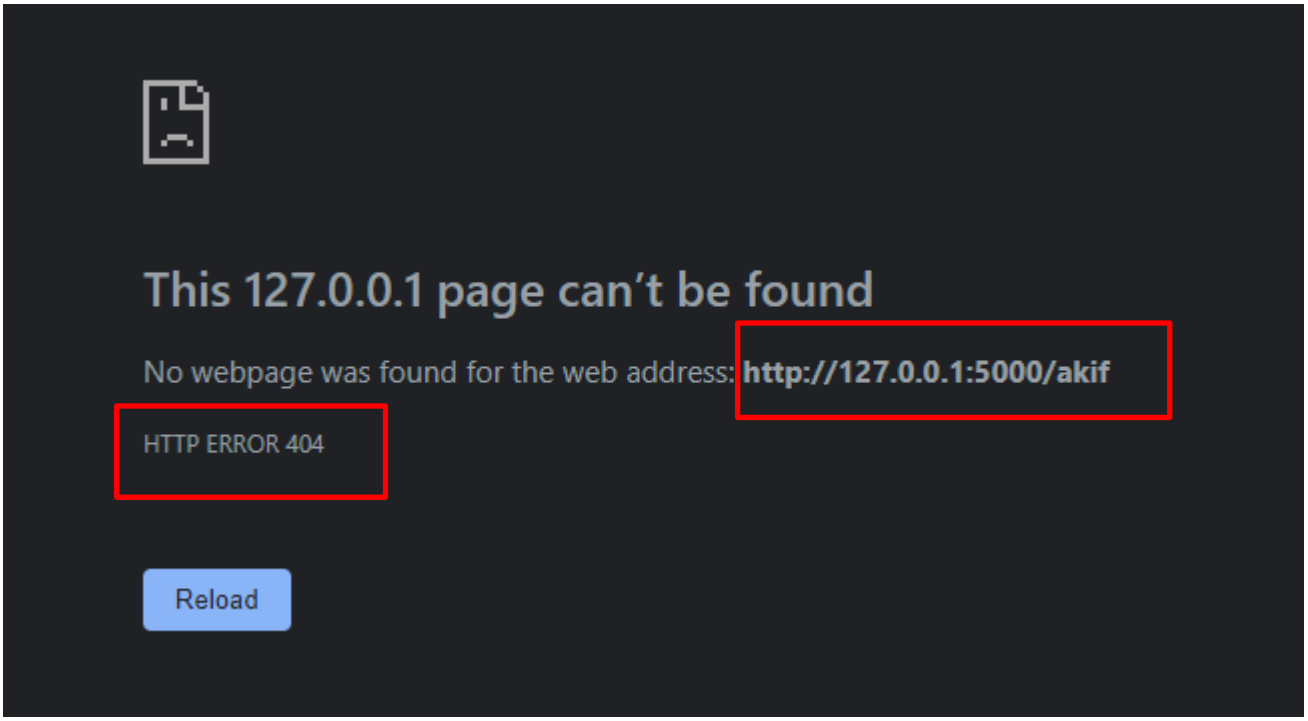
Connect



Hello World!

171044098 - Akif Kartal

Try to Connect wrong address



*As you can see, we got **error 404** as expected.

Assignment 2: UDP Pinger

Create Server Socket

```
serverSocket = socket(AF_INET, SOCK_DGRAM)
# Assign IP address and port number to socket
serverSocket.bind(('', 12000))
```

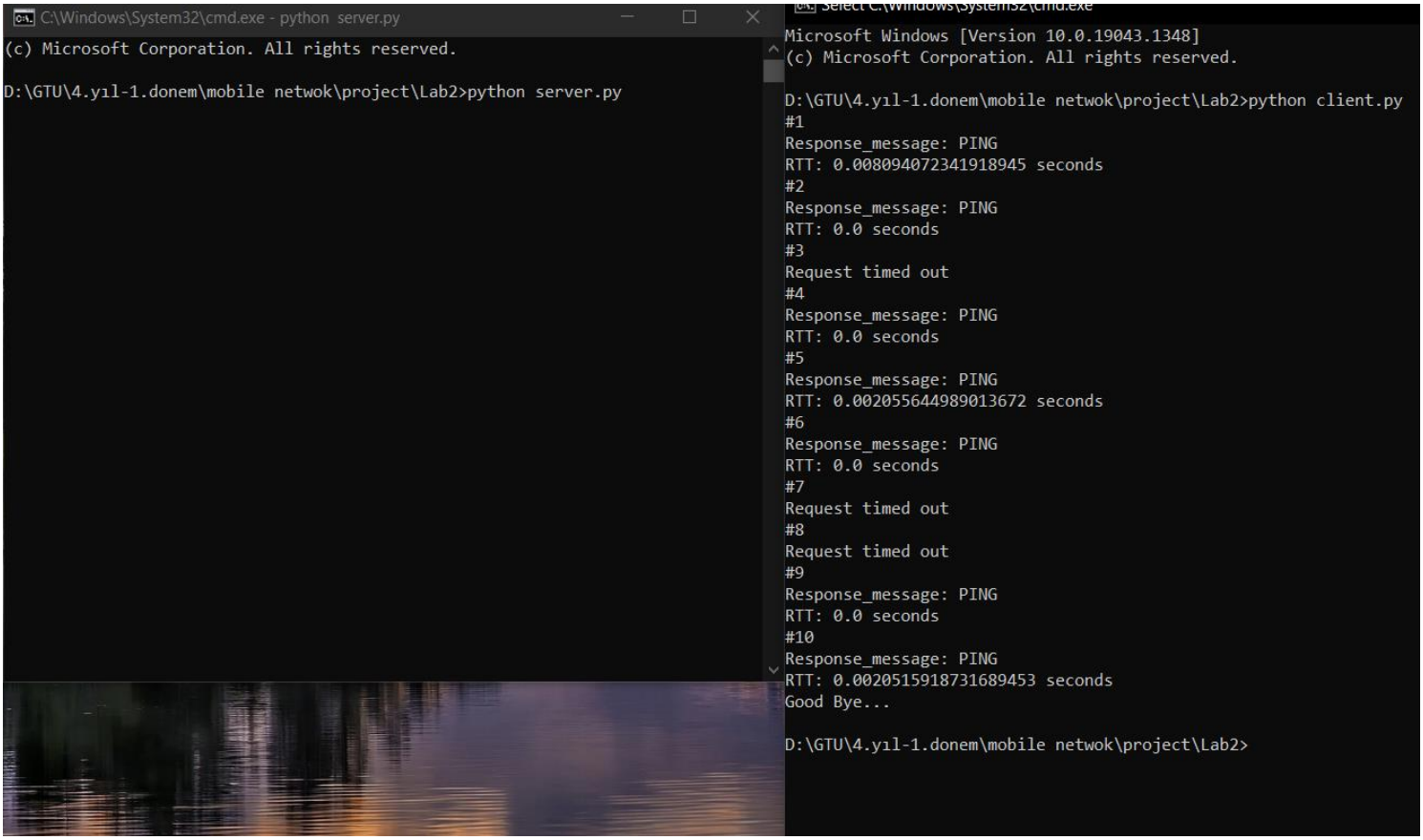
Create client socket and set message as “ping”

```
import time
from socket import *

client_socket = socket(AF_INET, SOCK_DGRAM)
client_socket.settimeout(1.0)
message = 'ping'
# server adress from given server code
address = ("localhost", 12000)
```

Communicate between client and server

```
for i in range(1, 11):
    start = time.time()
    client_socket.sendto(message.encode(), address)
    try:
        response, server = client_socket.recvfrom(1024)
        end = time.time()
        RTT = end - start

        print('#' + str(i))
        print('Response_message: ' + response.decode())
        print('RTT: ' + str(RTT) + " seconds")
    except timeout:
        print('#' + str(i))
        print('Request timed out')
print("Good Bye...")
client_socket.close()
```

Testing



Assignment 3: Mail Client

Setting Google Gmail Server and connecting

```
# Choose a mail server (e.g. Google mail server) and call it mailserver
mailserver = ('smtp.gmail.com', 587)  # Fill in start #Fill in end
# Fill in start
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect(mailserver)
# Fill in end
```

Setting Gmail Account and connect

In order to test, a Gmail account was created and some settings has been applied that account.

```
# Encrypt the socket
clientSocket = ssl.wrap_socket(clientSocket)
clientSocket.send("AUTH LOGIN ".encode() + b64encode("cse476.2022@gmail.com".encode()) + "\r\n".encode())
recv3 = clientSocket.recv(1024)
print(recv3)
if (recv3[:3] != "334"):
    print('334 reply not received from server.')

clientSocket.send(b64encode("Gtuforever123.".encode()) + "\r\n".encode())
recv4 = clientSocket.recv(1024)
print(recv4)
if (recv4[:3] != "235"):
    print('235 reply not received from server.')
```

e-mail

password

Setting MAIL FROM command

```
# Send MAIL FROM command and print server response.
# Fill in start
mailFrom = "MAIL FROM: <cse476.2022@gmail.com>\r\n"
clientSocket.send(mailFrom.encode())
recv5 = clientSocket.recv(1024)
print(recv5)
if (recv5[:3] != '250'):
    print('250 reply not received from server.')
# Fill in end
```

Setting RCPT TO command and print server response.

```
# Send RCPT TO command and print server response.
# Fill in start
rcptToCommand = "RCPT TO: <akif.kartal03@gmail.com>\r\n"
clientSocket.send(rcptToCommand.encode())
recv6 = clientSocket.recv(1024)
print(recv6)
if (recv6[:3] != '250'):
    print('250 reply not received from server.')
# Fill in end
```

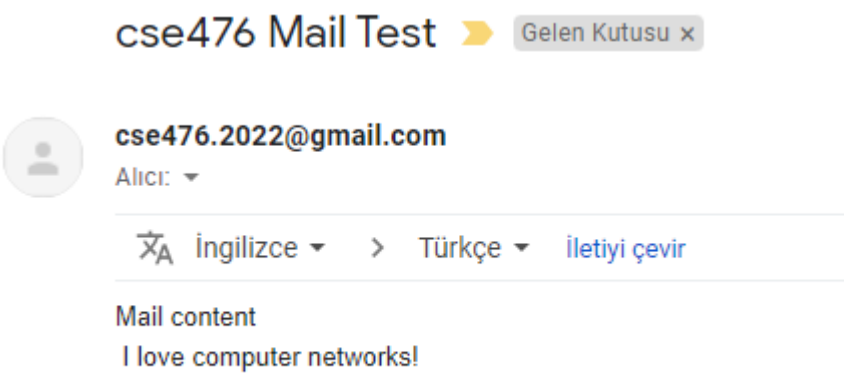
Sending message data.

```
# Send message data.
# Fill in start
message = "SUBJECT: cse476 Mail Test\nMail content" + msg + endmsg
clientSocket.send(message.encode())
recv8 = clientSocket.recv(1024)
print(recv8)
if (recv8[:3] != '250'):
    print('250 reply not received from server.')
# Fill in end
```

Output:

Mails was sent successfully.

Send to Gmail Account



Send to Outlook(GTU) Account

How to run all codes

Just run python files.