

Comp 2322 Computer Networking

Lab 4: Socket Programming

Due time: 11:59pm, March 25, 2024, Monday

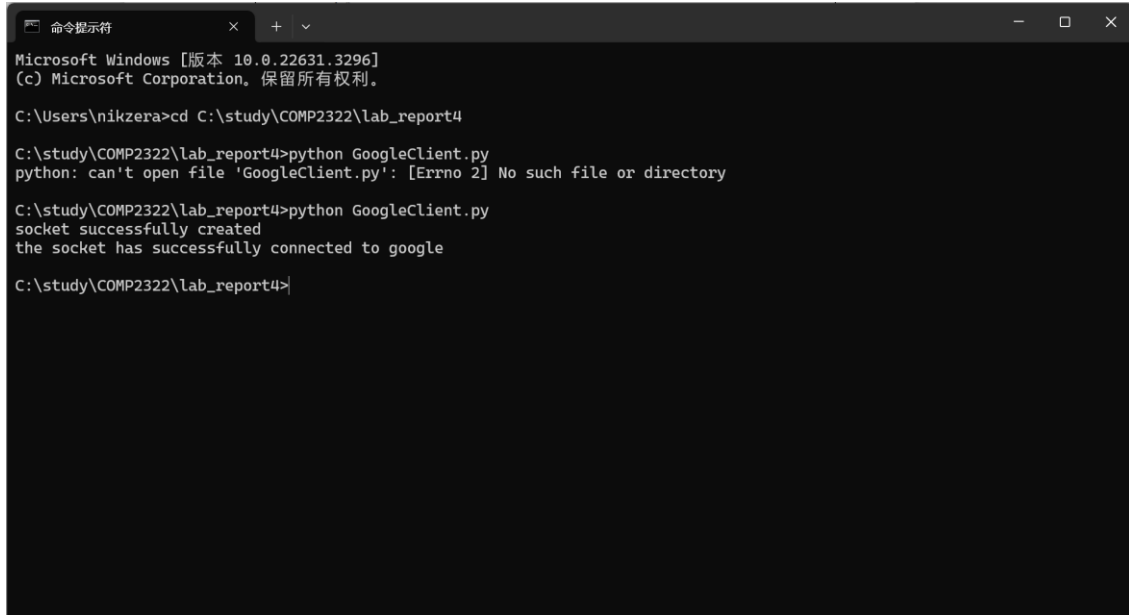
Total marks: 8 points

HE YIYANG

22100143D

Q1.

Running result:



```
命令提示符
Microsoft Windows [版本 10.0.22631.3296]
(c) Microsoft Corporation。保留所有权利。

C:\Users\nikzera>cd C:\study\COMP2322\lab_report4

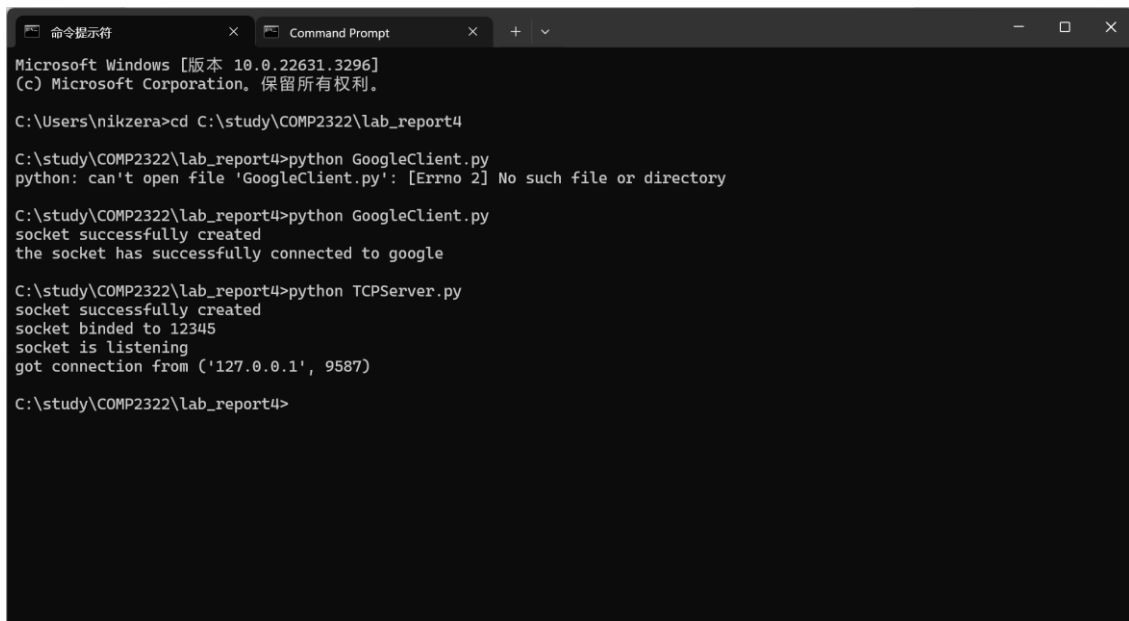
C:\study\COMP2322\lab_report4>python GoogleClient.py
python: can't open file 'GoogleClient.py': [Errno 2] No such file or directory

C:\study\COMP2322\lab_report4>python GoogleClient.py
socket successfully created
the socket has successfully connected to google

C:\study\COMP2322\lab_report4>
```

Q2.

Running result:



```
命令提示符
Microsoft Windows [版本 10.0.22631.3296]
(c) Microsoft Corporation。保留所有权利。

C:\Users\nikzera>cd C:\study\COMP2322\lab_report4

C:\study\COMP2322\lab_report4>python GoogleClient.py
python: can't open file 'GoogleClient.py': [Errno 2] No such file or directory

C:\study\COMP2322\lab_report4>python GoogleClient.py
socket successfully created
the socket has successfully connected to google

C:\study\COMP2322\lab_report4>python TCPServer.py
socket successfully created
socket binded to 12345
socket is listening
got connection from ('127.0.0.1', 9587)

C:\study\COMP2322\lab_report4>
```

```
命令提示符 x Command Prompt x + v
Microsoft Windows [版本 10.0.22631.3296]
(c) Microsoft Corporation. 保留所有权利。

C:\Users\nikzera>cd C:\study\COMP2322\lab_report4
C:\study\COMP2322\lab_report4>python TCPClient.py
C:\study\COMP2322\lab_report4>python TCPClient.py
C:\study\COMP2322\lab_report4>python TCPClient.py
from server: thank you for connecting
C:\study\COMP2322\lab_report4>
```

Q3.

Socket port number = 40143

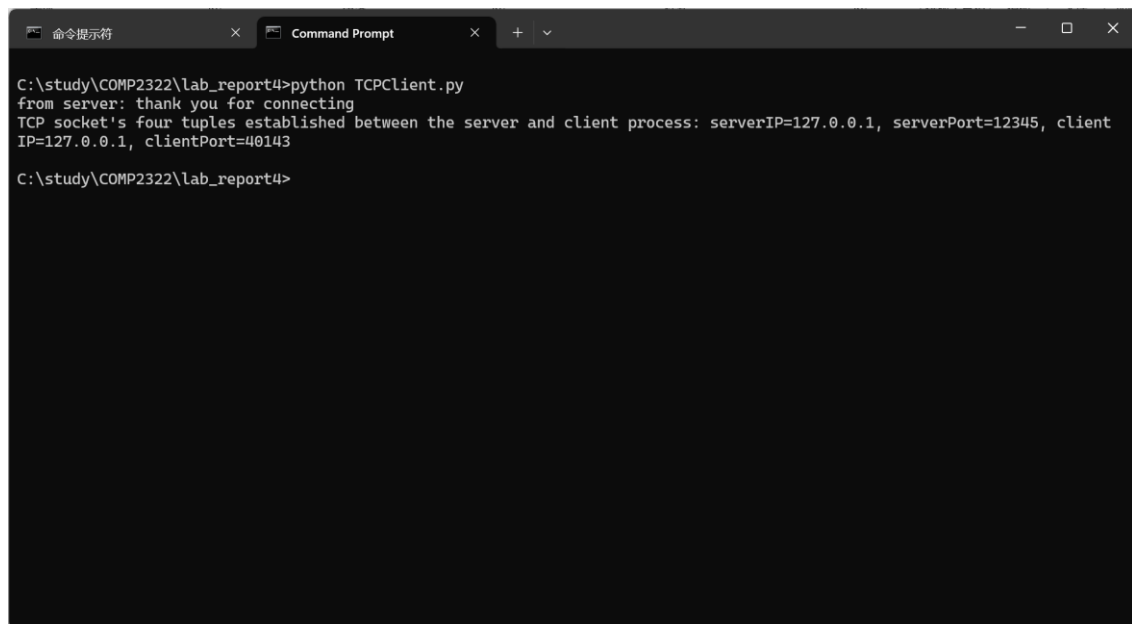
```
命令提示符 x Command Prompt x + v
C:\study\COMP2322\lab_report4>python TCPServer.py
socket successfully created
socket binded to 12345
socket is listening
got connection from ('127.0.0.1', 47845)

C:\study\COMP2322\lab_report4>python TCPServer.py
socket successfully created
socket binded to 12345
socket is listening
got connection from ('127.0.0.1', 47845)

C:\study\COMP2322\lab_report4>python TCPServer.py
socket successfully created
socket binded to 12345
socket is listening
got connection from ('127.0.0.1', 47845)

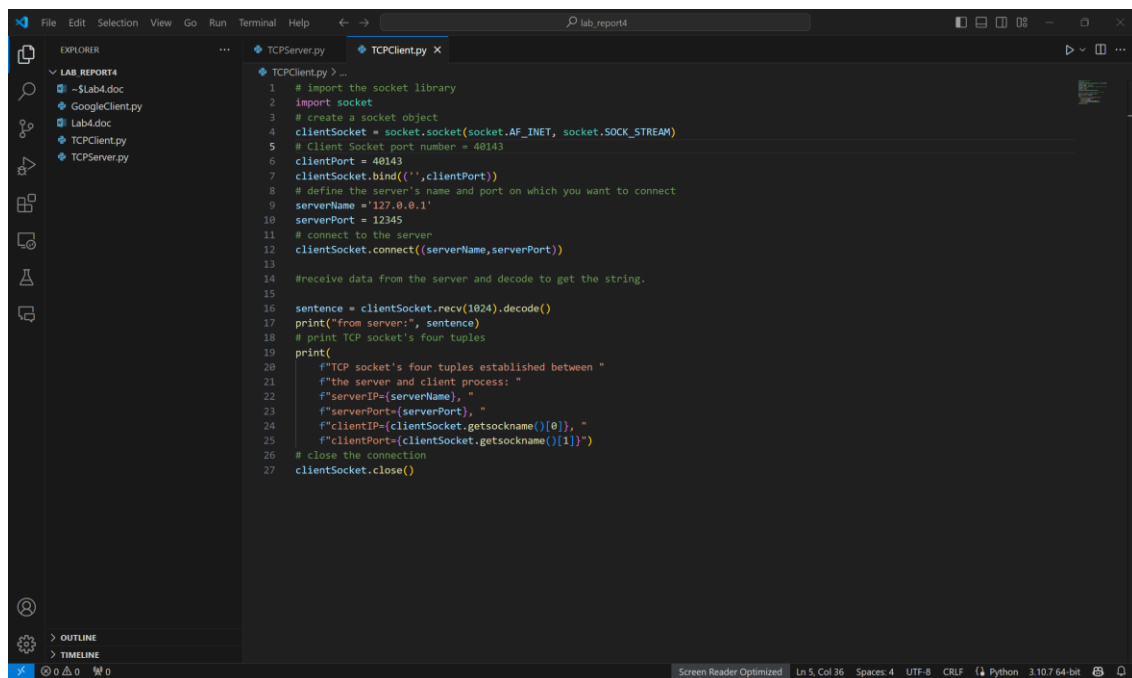
C:\study\COMP2322\lab_report4>python TCPServer.py
socket successfully created
socket binded to 12345
socket is listening
got connection from ('127.0.0.1', 40143)

C:\study\COMP2322\lab_report4>
```



```
命令提示符
C:\study\COMP2322\lab_report4>python TCPClient.py
from server: thank you for connecting
TCP socket's four tuples established between the server and client process: serverIP=127.0.0.1, serverPort=12345, client
IP=127.0.0.1, clientPort=40143

C:\study\COMP2322\lab_report4>
```



```
EXPLORER
LAB_REPORT4
- Lab4.doc
GoogleClient.py
Lab4.doc
TCPClient.py
TCPServer.py

TCPClient.py X
1 # import the socket library
2 import socket
3 # create a socket object
4 clientSocket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
5 # Client Socket port number = 40143
6 clientPort = 40143
7 clientSocket.bind(('',clientPort))
8 # define the server's name and port on which you want to connect
9 serverName = '127.0.0.1'
10 serverPort = 12345
11 # connect to the server
12 clientSocket.connect((serverName,serverPort))
13
14 #Receive data from the server and decode to get the string.
15
16 sentence = clientSocket.recv(1024).decode()
17 print("from server:", sentence)
18 # print TCP socket's four tuples
19 print(
20     f"TCP socket's four tuples established between "
21     f"the server and client process: "
22     f"serverIP={serverName}, "
23     f"serverPort={serverPort}, "
24     f"clientIP={clientSocket.getsockname()[0]}, "
25     f"clientPort={clientSocket.getsockname()[1]}")
26 # close the connection
27 clientSocket.close()
```

Q4.

Password = 00143

Correct:

```
命令提示符  Command Prompt
socket binded to 12345
socket is listening
got connection from ('127.0.0.1', 47845)

C:\study\COMP2322\lab_report4>python TCPServer.py
socket successfully created
socket binded to 12345
socket is listening
got connection from ('127.0.0.1', 47845)

C:\study\COMP2322\lab_report4>python TCPServer.py
socket successfully created
socket binded to 12345
socket is listening
got connection from ('127.0.0.1', 47845)

C:\study\COMP2322\lab_report4>python TCPServer.py
socket successfully created
socket binded to 12345
socket is listening
got connection from ('127.0.0.1', 40143)

C:\study\COMP2322\lab_report4>python TCPServer.py
socket successfully created
socket binded to 12345
socket is listening
got connection from ('127.0.0.1', 13376)
from client: 00143

C:\study\COMP2322\lab_report4>

命令提示符  Command Prompt
C:\study\COMP2322\lab_report4>python TCPClient.py
from server: thank you for connecting
TCP socket's four tuples established between the server and client process: serverIP=127.0.0.1, serverPort=12345, client
IP=127.0.0.1, clientPort=40143

C:\study\COMP2322\lab_report4>python TCPClient.py
Enter the password:00143
from server: Your password is correct!

C:\study\COMP2322\lab_report4>
```

Incorrect:

```
命令提示符  Command Prompt
socket is listening
got connection from ('127.0.0.1', 47845)

C:\study\COMP2322\lab_report4>python TCPServer.py
socket successfully created
socket binded to 12345
socket is listening
got connection from ('127.0.0.1', 47845)

C:\study\COMP2322\lab_report4>python TCPServer.py
socket successfully created
socket binded to 12345
socket is listening
got connection from ('127.0.0.1', 40143)

C:\study\COMP2322\lab_report4>python TCPServer.py
socket successfully created
socket binded to 12345
socket is listening
got connection from ('127.0.0.1', 13376)
from client: 00143

C:\study\COMP2322\lab_report4>python TCPServer.py
socket successfully created
socket binded to 12345
socket is listening
got connection from ('127.0.0.1', 13462)
from client: 12345

C:\study\COMP2322\lab_report4>

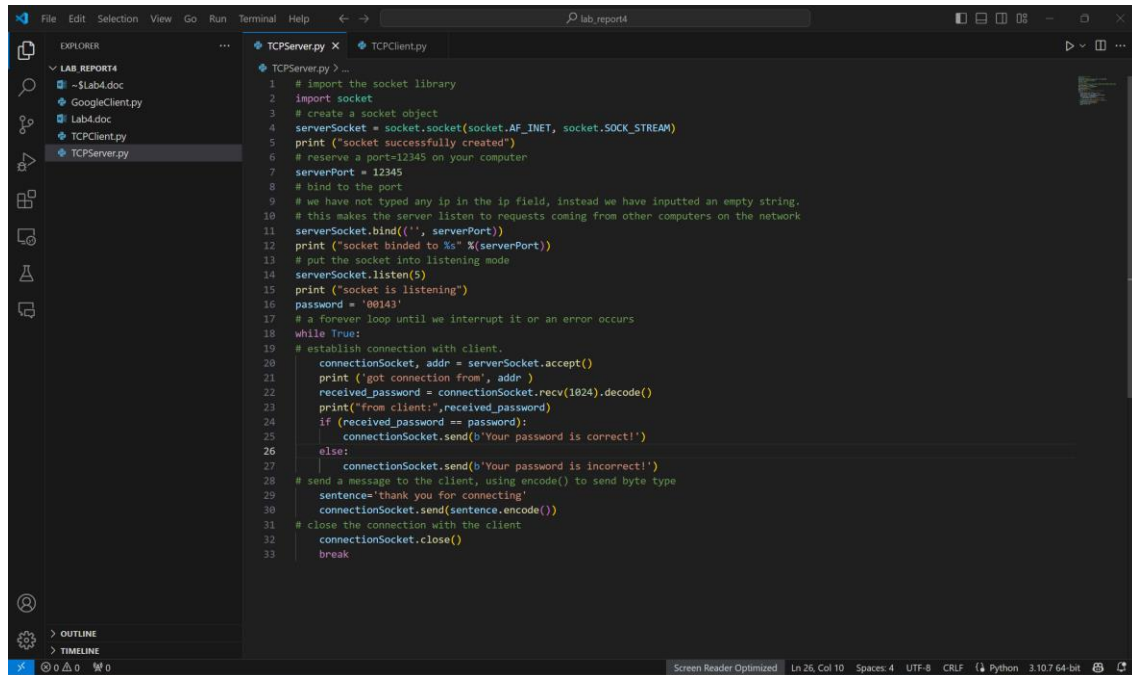
命令提示符  Command Prompt
C:\study\COMP2322\lab_report4>python TCPClient.py
from server: thank you for connecting
TCP socket's four tuples established between the server and client process: serverIP=127.0.0.1, serverPort=12345, client
IP=127.0.0.1, clientPort=40143

C:\study\COMP2322\lab_report4>python TCPClient.py
Enter the password:00143
from server: Your password is correct!

C:\study\COMP2322\lab_report4>python TCPClient.py
Enter the password:12345
from server: Your password is incorrect!

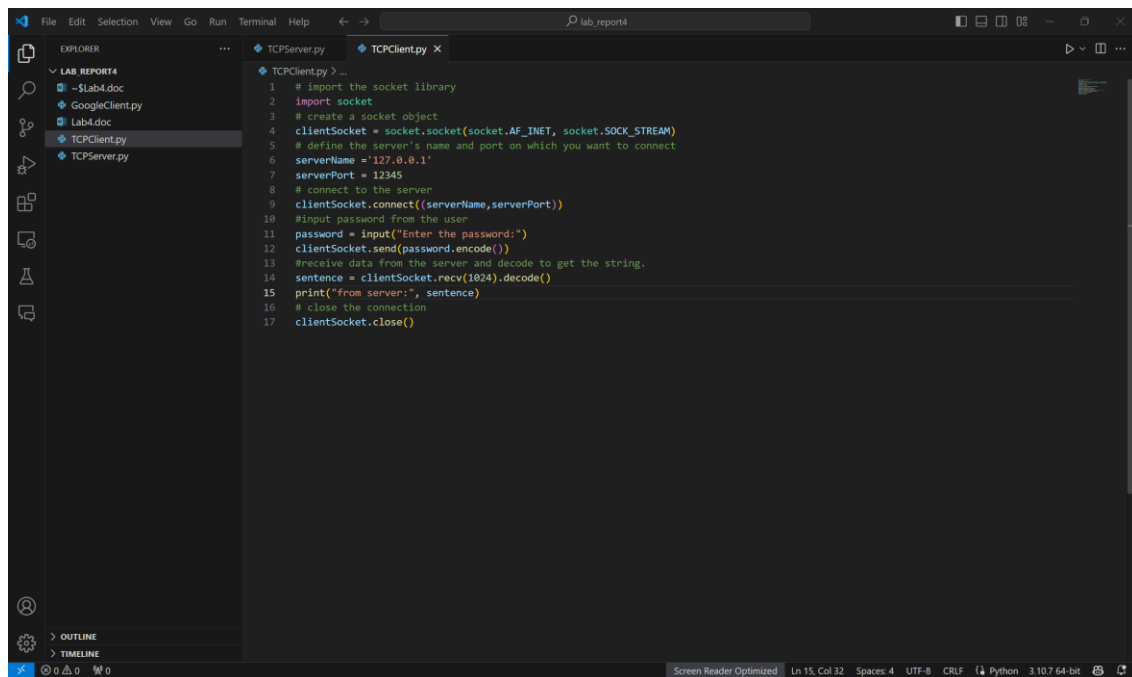
C:\study\COMP2322\lab_report4>
```

Code:



This screenshot shows the Visual Studio Code editor with the file explorer on the left displaying a project named 'LAB_REPORT4'. The files listed are '~\$Lab4.doc', 'GoogleClient.py', 'Lab4.doc', 'TCPClient.py', and 'TCPServer.py'. The 'TCPServer.py' file is selected and its code is displayed in the main editor window. The code is a Python script that implements a simple TCP server. It imports the 'socket' library, creates a 'serverSocket' object, binds it to an empty string and port 12345, and starts listening for connections. When a connection is accepted, it receives data from the client, decodes it, and checks if it matches the password '100143'. If the password is correct, it sends a confirmation message; otherwise, it sends an incorrect password message and then sends a 'thank you for connecting' message before closing the connection. The server runs in a continuous loop until interrupted.

```
1 # import the socket library
2 import socket
3 # create a socket object
4 serverSocket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
5 print ("socket successfully created")
6 # reserve a port-12345 on your computer
7 serverPort = 12345
8 # bind to the port
9 # we have not typed any ip in the ip field, instead we have inputted an empty string.
10 # this makes the server listen to requests coming from other computers on the network
11 serverSocket.bind(('', serverPort))
12 print ("socket binded to %s" %(serverPort))
13 # put the socket into listening mode
14 serverSocket.listen(5)
15 print ("socket is listening")
16 password = '100143'
17 # a forever loop until we interrupt it or an error occurs
18 while True:
19     # establish connection with client.
20     connectionSocket, addr = serverSocket.accept()
21     print ('got connection from', addr )
22     received_password = connectionSocket.recv(1024).decode()
23     print("from client:",received_password)
24     if (received_password == password):
25         connectionSocket.send(b'Your password is correct!')
26     else:
27         connectionSocket.send(b'Your password is incorrect!')
28     # send a message to the client, using encode() to send byte type
29     sentence='thank you for connecting'
30     connectionSocket.send(sentence.encode())
31 # close the connection with the client
32 connectionSocket.close()
33 break
```



This screenshot shows the Visual Studio Code editor with the file explorer on the left displaying the same 'LAB_REPORT4' project. The 'TCPClient.py' file is now selected, and its code is shown in the main editor window. This script implements a simple TCP client. It imports the 'socket' library, creates a 'clientSocket' object, and connects it to the server at IP '127.0.0.1' on port 12345. The client prompts the user to enter a password, encodes it, and sends it to the server. It then receives data from the server, decodes it, and prints the received message. Finally, it closes the connection. The status bar at the bottom indicates the cursor is at line 15, column 32.

```
1 # import the socket library
2 import socket
3 # create a socket object
4 clientSocket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
5 # define the server's name and port on which you want to connect
6 serverName = '127.0.0.1'
7 serverPort = 12345
8 # connect to the server
9 clientSocket.connect((serverName,serverPort))
10 #input password from the user
11 password = input("Enter the password:")
12 clientSocket.send(password.encode())
13 #receive data from the server and decode to get the string.
14 sentence = clientSocket.recv(1024).decode()
15 print("from server:", sentence)
16 # close the connection
17 clientSocket.close()
```