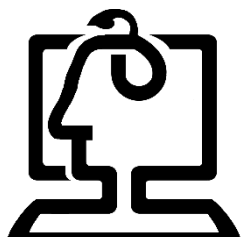


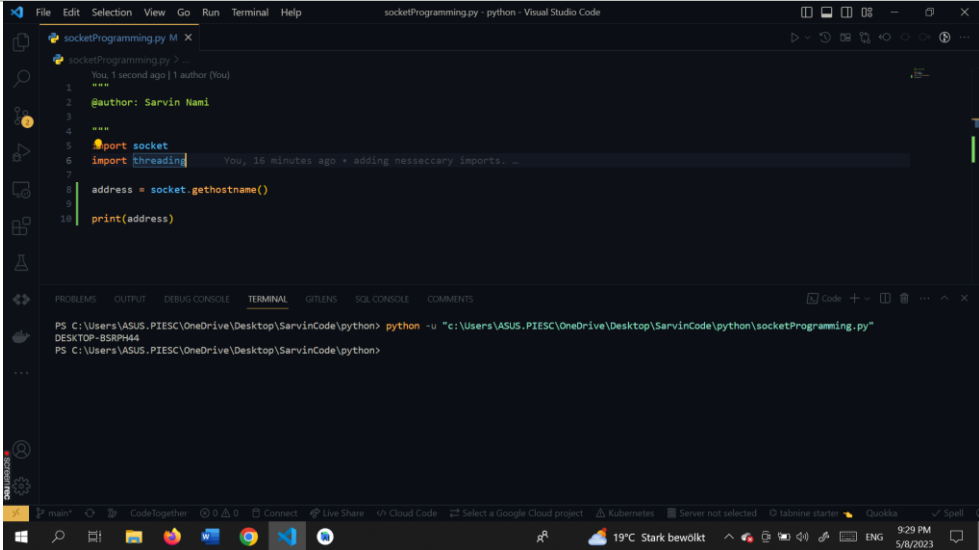


دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

فرم گزارش کار آزمایشگاه شبکه



دانشکده مهندسی کامپیوتر

| | | | | | |
|--------------------|--|----------------|---------|--------------------|---|
| نام و نام خانوادگی | سروین نامی | شماره دانشجویی | 9931103 | نام و شماره آزمایش | 6 |
| هدف آزمایش | برقراری ارتباط tcp بین کلاینت و سرور نوشته شده. | | | | |
| ابزارهای مورد نیاز | لپ تاپ مجهز به پایتون | | | | |
| شرح آزمایش | <div></div> | | | | |
| | در ابتدا نام دستگاه را چاپ کردیم. | | | | |

The screenshot shows a Visual Studio Code window with a file named `socketProgramming.py` open. The code in the file is as follows:

```
1  """
2  @author: Sarvin Nami
3  """
4  from socket import *
5  import threading
6
7  address = gethostbyname(gethostname())
8
9  print(address)
```

The terminal at the bottom shows the command `python -u "c:\Users\ASUS.PIESC\OneDrive\Desktop\SarvinCode\python\socketProgramming.py"` being executed, which outputs the IP address `192.168.3.7`.

آدرس ip دستگاه من چاپ شد.

نتایج سمت سرور:

The screenshot shows two files, `server.py` and `client.py`, open in Visual Studio Code. The `server.py` file contains the following code:

```
15 def start(server):
16     print("Server is starting...")
17     server.listen()
18     while True:
19         connection, address = server.accept()
20         thread = Thread(target = clientHandler, args = (connection, address))
21         thread.start()
22
23 def clientHandler(connection, address):
24     print(f"Client from IP address {address} successfully connected")
25     connected = True
26     while connected:
27         messageLen = int(connection.recv(MESSAGE_SIZE).decode(ENCODING))
28         msg = connection.recv(messageLen).decode(ENCODING)
```

The `client.py` file contains the following code:

```
4  """
5  from socket import *
6
7  PORT, MESSAGE_SIZE, ENCODING = 7447, 64, "utf-8"
8
9  def main():
10     clientSocket = socket(AF_INET, SOCK_STREAM)
11     address = gethostbyname(gethostname())
12     SERVER_info = (address, PORT)
13     clientSocket.connect(SERVER_info)
14     print("Client is up!")
15     sendMsg(clientSocket, "Hey Meow to everyone! ^-^")
16     sendMsg(clientSocket, "DISCONNECT")
17
18 def sendMsg(client, msg):
```

The terminal at the bottom shows the command `python -u "c:\Users\ASUS.PIESC\OneDrive\Desktop\SarvinCode\python\socketProgramming\server.py"` being executed, which outputs the following messages:

```
Server is starting...
Client from IP address ('192.168.3.7', 7226) successfully connected! ^-^New Message:
Hey Meow to everyone! ^-^
New Message:
DISCONNECT
```

نتایج سمت کلاینت:

```
server.py
15
16 def start(server):
17     print("Server is starting...")
18     server.listen()
19     while True:
20         connection, address = server.accept()
21         thread = Thread(target = clientHandler, args = (connection, address))
22         thread.start()
23
24 def clientHandler(connection, address):
25     print(f"Client from IP address {address} successfully connected")
26     connected = True
27     while connected:
28         messageLen = int(connection.recv(MESSAGE_SIZE).decode(ENCODING))
29         msg = connection.recv(messageLen).decode(ENCODING)
30
31 client.py
32
33 """
34 from socket import *
35
36 PORT, MESSAGE_SIZE, ENCODING = 7447, 64, "utf-8"
37
38 def main():
39     clientSocket = socket(AF_INET, SOCK_STREAM)
40     address = gethostbyname(gethostname())
41     SERVER_info = (address, PORT)
42     clientSocket.connect(SERVER_info)
43     print("Client is up!")
44     sendMsg(clientSocket, "Hey Meow to everyone! ^-^")
45     sendMsg(clientSocket, "DISCONNECT")
46
47 def sendMsg(client, msg):
```

```
PS C:\Users\ASUS.PIEESC\OneDrive\Desktop\SarvinCode\python> python -u "c:\Users\ASUS.PIEESC\OneDrive\Desktop\SarvinCode\python\socketProgramming\client.py"
client is up!
```

کد سمت سرور با کامنت برای توضیحات خط به خط:

```
"""
@author: Sarvin Nami

"""
from socket import *
from threading import *
# defining port number, our tcp message size and our encoding mode.
PORT, MESSAGE_SIZE, ENCODING = 7447, 64, "utf-8"

def main():
    # the ip address of my laptop as server
    address = gethostbyname(gethostname())
    # defining host information and server socket
    HOST_info, serverSocket = (address, PORT), socket(AF_INET,
SOCK_STREAM)
    # binding the server socket via host info
    serverSocket.bind(HOST_info)
    # starting the server ^-^
    start(serverSocket)

def start(server):
    print("Server is starting...")
    # server is waiting for clients
    server.listen()
    while True:
        # every moment server is ready to accept connections from
clients
```

```

        connection, address = server.accept()
        # having a thread for each client
        thread = Thread(target = clientHandler, args = (connection,
address))
        thread.start()

# thread function
def clientHandler(connection, address):
    print(f"Client from IP address {address} successfully connected! ^-
^")
    connected = True
    while connected:
        # figuring out the length of out message
        messageLen =
int(connection.recv(MESSAGE_SIZE).decode(ENCODING))
        # reading the message to reach the defined length
        msg = connection.recv(messageLen).decode(ENCODING)
        print(f"New Message:\n{msg}")
        # disconnect time T-T
        if msg == "DISCONNECT":
            connected = False
        # closing the connection
        connection.close()

if __name__ == "__main__":
    main()

```

کد سمت کلاینت با کامنت:

```

"""
@author: Sarvin Nami
"""
from socket import *

# defining basic setting as server
PORT, MESSAGE_SIZE, ENCODING = 7447, 64, "utf-8"

def main():
    # defining client socket
    clientSocket = socket(AF_INET, SOCK_STREAM)
    # the ip address of my laptop as client
    address = gethostbyname(gethostname())
    # defining server information
    SERVER_info = (address, PORT)
    # connecting client socket to the server
    clientSocket.connect(SERVER_info)

```

```

print("Client is up!")
# sending messages and ending the connection
sendMsg(clientSocket, "Hey Meow to everyone! ^-^")
sendMsg(clientSocket, "DISCONNECT")

def sendMsg(client, msg):
    # encoding the message
    message = msg.encode(ENCODING)
    # calculating message length, encoding it, turning it to bytes and
    sending it.
    msgLen = str(len(message)).encode(ENCODING)
    msgLen += b' ' * (MESSAGE_SIZE - len(msgLen))
    client.send(msgLen)
    # sending the message
    client.send(message)

if __name__ == "__main__":
    main()

```

به راحتی میتوان انواع ارتباطات برنامه های کاربردی را برای انواع انقالات tcp و udp توسعه داد.

نتیجه-
گیری