

BT18CSE044

In video all the things explained properly ,
here I'm adding snippets and some useful
instructions and the features

CLIENT.PY

- 1 . Input server ip and port number
2. Press q , "Q" , "Quit" to exit
3. Press y to run again and n to come out and thus closing it .
4. All kinds of exceptions handled here

SERVER IN OTHER SLIDES (STARTING FROM PAGE 2)

SERVER 1(handling one at a time(client req))

The screenshot shows the Visual Studio Code interface with a Python file named `server4.py` open. The code is a simple server that listens on port 83 and handles client requests one at a time. The terminal output shows the server running and handling a client request.

```
server4.py
51 connection, client_address = s.accept()
52 print('connection from', client_address, file=sys.stderr)
53 connection.setblocking(0)
54 inputs.append(connection)
55 # Give the connection a queue for data
56 # we want to send
57 message_queues[connection] = queue.Queue()
58
59 else:
60 data = s.recv(1024)
61 if data:
62     # A readable client socket has data
63     print(' received {} from {}'.format(data, s.getpeername()), file=sys.stderr,)
64     message_queues[s].put(data)
65     # Add output channel for response
66     if s not in outputs:
67         outputs.append(s)
```

Terminal Output:

```
PS C:\Users\KETAN SARODE\Desktop\cn_assign> python s
erver1.py 83
socket binded to port 83
socket is listening
Connected to : 127.0.0.1 : 54440
You gave me the equation: 9/0
Dye
socket binded to port 83
socket is listening
Connected to : 127.0.0.1 : 54445
You gave me the equation: 3+99
[]
```

For RUNNING

For Server side

```
>> server1.py 83
```

For Client

```
>> client.py 127.0.0.1 83
```

Basically working as per Question , handling all the exceptions and handling all the OS , Index , Value , Connection errors and other terminal errors , handling keyboard interrupt.

SERVER 2(Threading – Multi threaded server)

```
server2.py
55 # establish connection with client
56 c, addr = s.accept()
57 print('Connected to :', addr[0], ':', addr[1])
58 start_new_thread(client_thread, (c, ))
59
60
61
62 if __name__ == '__main__':
63
64
65     try:
66         host = socket.gethostname("localhost") # Get local machine name
67         port = int(sys.argv[1])
68     except IndexError:
69         print("Wrong arguments - Terminating the Program")
70         exit(1)
71     ThreadCount = 0 #for taking the count of respective threads
72
73     s = socket.socket(socket.AF_INET, socket.SOCK_STREAM) #TCP/IP Socket Creation
74     binding(s) #binding to the respective port
75     listing(s,5) #put the socket into listening mode
```

TERMINAL

Try the new cross-platform PowerShell https://aka.ms/powershell

python s

server2.py 82

socket binded to port 82

Connected to : 127.0.0.1 : 53522

Thread Number: 1

You gave me the equation: 2+7

Connected to : 127.0.0.1 : 53523

Thread Number: 2

You gave me the equation: 7+5

You gave me the equation: 9+11

Bye closing...

You gave me the equation: 9+88

Do you want to continue(y/n) : y

Please give me your equation (Ex: 2+2) or Q to quit: 2+7

The answer is: 9

Do you want to continue(y/n) : n

PS C:\Users\KETAN SARODE\Desktop\cn_assign>

python clie

PS C:\Users\KETAN SARODE\Desktop\cn_assign> python client.py 127.0.0.1 82

The IP address of the server is: 127.0.0.1

The port number of the server is: 82

Please give me your equation (Ex: 2+2) or Q to quit: 2+7

The answer is: 9

Do you want to continue(y/n) : y

Please give me your equation (Ex: 2+2) or Q to quit: 9+88

The answer is: 97

Do you want to continue(y/n) :

For RUNNING

For Server side

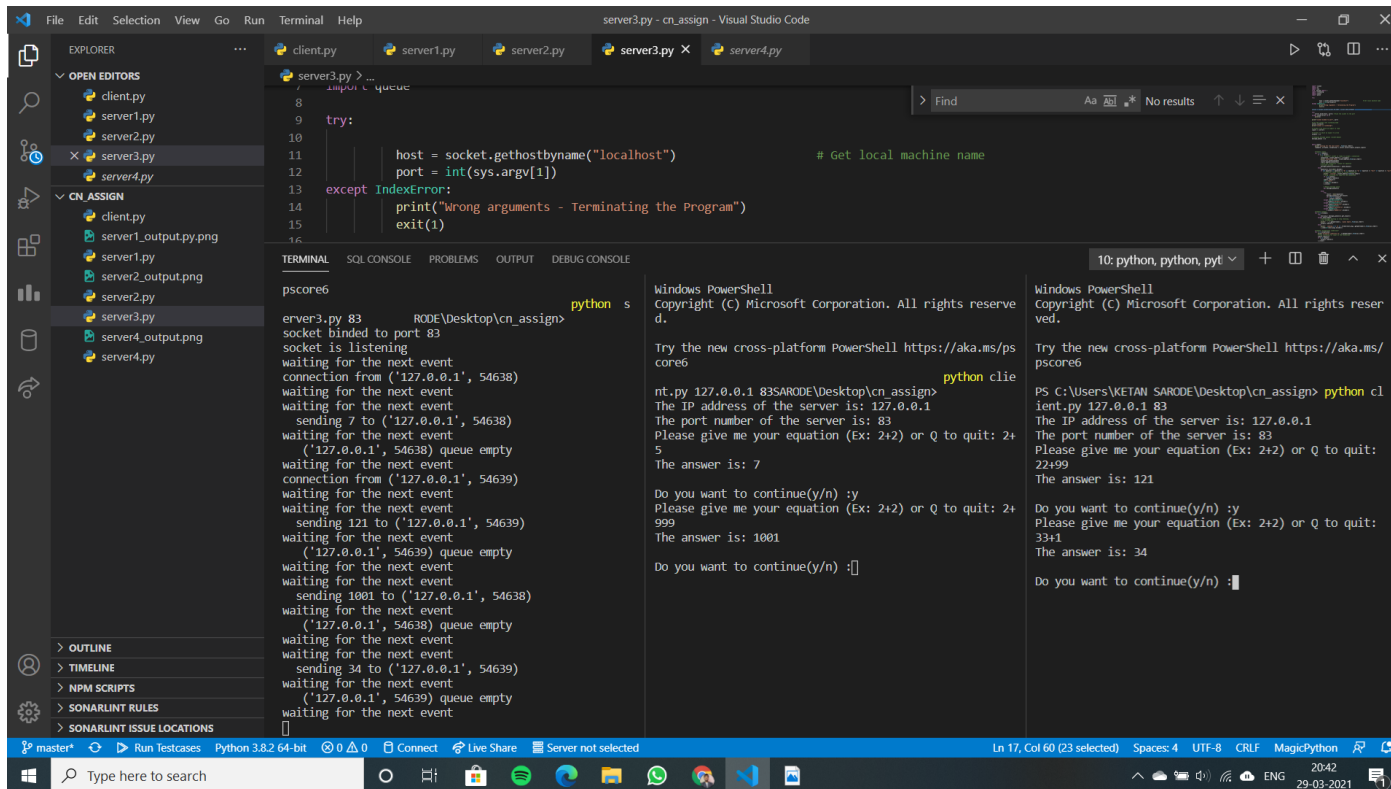
>> server2.py 83

For Client

>> client.py 127.0.0.1 83

Basically working as per Question , handling all the exceptions and handling all the OS , Index , Value , Connection errors and other terminal errors , handling keyboard interrupt , supports multiple clients , threading implemented.

SERVER 3(using select())



For RUNNING

For Server side

>> server3.py 83

For Client

>> client.py 127.0.0.1 83

Basically working as per Question , handling all the exceptions and handling all the OS , Index , Value , Connection errors and other terminal errors , handling keyboard interrupt , supports multiple clients , select() method implemented.

SERVER 4

The screenshot displays the Visual Studio Code interface with a project named 'cn_assign'. The Explorer sidebar on the left shows the file structure with 'server4.py' selected. The main editor window shows the code for 'server4.py', which uses a queue to handle multiple client connections. The code includes comments like '# Give the connection a queue for data' and '# we want to send'. The Terminal panel at the bottom shows the execution of 'python server4.py' in a PowerShell window. The output shows the server listening on port 81 and handling multiple client connections from '127.0.0.1'. The client sends messages like 'I am Ketan' and '33+21', and the server responds with the IP address, port number, and the result of the calculation. The status bar at the bottom indicates the current file is 'server4.py' and the Python interpreter is 'Python 3.8.2 64-bit'.

```
server4.py ...
54 inputs.append(connection)
55 # Give the connection a queue for data
56 # we want to send
57 message_queues[connection] = queue.Queue()
58
59 else:
60 data = s.recv(1024)
61 if data:
```

```
PS C:\Users\KETAN SARODE\Desktop\cn_assign> python server4.py
socket is listening
waiting for the next event
connection from ('127.0.0.1', 52854)
waiting for the next event
received b'2+4' from ('127.0.0.1', 52854)
waiting for the next event
sending b'2+4' to ('127.0.0.1', 52854)
waiting for the next event
('127.0.0.1', 52854) queue empty
waiting for the next event
connection from ('127.0.0.1', 52855)
waiting for the next event
received b'33+21' from ('127.0.0.1', 52855)
waiting for the next event
sending b'33+21' to ('127.0.0.1', 52855)
waiting for the next event
('127.0.0.1', 52855) queue empty
waiting for the next event
received b'33+45' from ('127.0.0.1', 52854)
waiting for the next event
sending b'33+45' to ('127.0.0.1', 52854)
waiting for the next event
('127.0.0.1', 52854) queue empty
waiting for the next event
received b'88+100' from ('127.0.0.1', 52855)
waiting for the next event
sending b'88+100' to ('127.0.0.1', 52855)
waiting for the next event
('127.0.0.1', 52855) queue empty
```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\KETAN SARODE\Desktop\cn_assign> python client.py 127.0.0.1 81
Please enter the message: I am Ketan
('127.0.0.1', 52685): sending b'I am Ketan'
PS C:\Users\KETAN SARODE\Desktop\cn_assign> python client.py 127.0.0.1 81
The IP address of the server is: 127.0.0.1
The port number of the server is: 81
Please give me your equation (Ex: 2+2) or Q to quit: 2+4
The answer is: 2+4
Do you want to continue(y/n) : y
Please give me your equation (Ex: 2+2) or Q to quit: 33+45
The answer is: 33+45
Do you want to continue(y/n) : n
PS C:\Users\KETAN SARODE\Desktop\cn_assign>
```

For RUNNING

For Server side

>> server4.py 83

For Client

>> client.py 127.0.0.1 83

Basically working as per Question , handling all the exceptions and handling all the OS , Index , Value , Connection errors and other terminal errors , handling keyboard interrupt , supports multiple clients , select() method implemented.