KHUSHI PRASAD
2001CS38
COMPUTER NETWORKS LABORATORY
ASSIGNMENT - 3

SERVER - 1:

server1.py : Your server program "server1.py" will be a single process server that can handle only one client at a time. If a second client tries to chat with the server while some other client's session is already in progress, the second client's socket operations should see an error (Display, "Client is already connected, please try after sometime."). After the first client closes the connection, the server should then accept connections from the other client.

Running server 1 and then connecting the client to it.

CASE – I:
In this case, after the client has connected to the server, they type in basic arithmetic equations. The server receives them, solves them and sends back the answer.

CLIENT:

```
uneednotknow@uneednotknow-VirtualBox:~/tut3$ python3 client.py 127.0.0.1 5000
Connected to server.
Input :3 + 4
Server replied:  7
Type the expression if you still want to continue. Type N to exit.
Input : 3 * 4
Server replied:  12
Type the expression if you still want to continue. Type N to exit.
Input :4 - 3
Server replied:  1
Type the expression if you still want to continue. Type N to exit.
Input : 4 / 2
Server replied:  2.0
Type the expression if you still want to continue. Type N to exit.
Input :
```

SERVER:

```
uneednotknow@uneednotknow-VirtualBox:~/tut3$ python3 server1.py 127.0.0.1 5000
Socket binded to 5000.
Socket is listening.
Connected to client:  ('127.0.0.1', 45854)
Connection closed by client.
Connected to client:  ('127.0.0.1', 45856)
Client sent:  3 + 4
Sending:  7
Client sent:   3 * 4
Sending:  12
Client sent:  4 - 3
Sending:  1
Client sent:   4 / 2
Sending:  2.0
```

CASE – II
In this case, the client sends some wrong inputs. The server distinguish the wrong inputs and sends back an appropriate error message:

CLIENT SIDE:

```
Input :3
Server replied:  Incorrect syntax.
Type the expression if you still want to continue. Type N to exit.
Input :3 % 4
Server replied:  Invalid operator.
Type the expression if you still want to continue. Type N to exit.
Input :3+4
Server replied:  Incorrect syntax.
Type the expression if you still want to continue. Type N to exit.
Input :
```

SERVER SIDE:

```
Client sent:  3
Sending:  Incorrect syntax.
Client sent:  3 % 4
Sending:  Invalid operator.
Client sent:  3+4
Sending:  Incorrect syntax.
```

CASE – III
In this case, we try to connect to the port already in use.
The connection fails, and a error message is displayed:

```
uneednotknow@uneednotknow-VirtualBox:~/tut3$ python3 server1.py 127.0.0.1 5000
Port is engaged right now:  [Errno 98] Address already in use
uneednotknow@uneednotknow-VirtualBox:~/tut3$
```

CASE – IV
In this case, a second client attempts to connect to the server when the server is already connected to a client.
The connection fails and the following error message is displayed:

```
uneednotknow@uneednotknow-VirtualBox:~/tut3$ python3 client.py 127.0.0.1 5000
Can't connect to server right now:  [Errno 110] Connection timed out
uneednotknow@uneednotknow-VirtualBox:~/tut3$
```

CASE – V
The client is asked if they want to continue or not. If not, they are prompted to enter N.
At the client side, the connection is broken.
At the server side, it is ready to accept new connection requests.

CLIENT SIDE:

```
Type the expression if you still want to continue. Type N to exit.
Input :N
Session Over.
uneednotknow@uneednotknow-VirtualBox:~/tut3$
```

SERVER SIDE:

```
Connection closed by client.
```

CASE - VI

A new client attempts connection after the previous client has disconnected. The connection request is accepted by the server.

CLIENT SIDE:

```
uneednotknow@uneednotknow-VirtualBox:~/tut3$ python3 client.py 127.0.0.1 5000
Connected to server.
Input : 4 + 5
Server replied:  9
Type the expression if you still want to continue. Type N to exit.
Input :N
Session Over.
uneednotknow@uneednotknow-VirtualBox:~/tut3$
```

SERVER SIDE:

```
Connected to client:  ('127.0.0.1', 46754)
Connection closed by client.
Connected to client:  ('127.0.0.1', 46756)
Client sent:   4 + 5
Sending:  9
Connection closed by client.
```

SERVER - 2

server2.py : Your server program "server2.py" will be a multi-threaded server that will create a new thread for every new client request it receives. Multiple clients should be able to simultaneously chat with the server.

Running server 2 and then connecting the client to it.

```
uneednotknow@uneednotknow-VirtualBox:~/tut3$ python3 server2.py 127.0.0.1 5000
Socket binded to 5000.
Server is listening.
```

CASE - I:

In this case, after the client has connected to the server, they type in basic arithmetic equations. The server receives them, checks if they are syntactically correct, solves them and sends back the answer or corresponding error message.

CLIENT:

```
uneednotknow@uneednotknow-VirtualBox:~/tut3$ python3 client.py 127.0.0.1 5000
Connected to server.
Input : 3 + 4
Server replied:  7
Type the expression if you still want to continue. Type N to exit.
Input : 2
Server replied:  Incoorect Syntax.
Type the expression if you still want to continue. Type N to exit.
Input :2
Server replied:  Incoorect Syntax.
Type the expression if you still want to continue. Type N to exit.
Input :2 * 6
Server replied:  12
Type the expression if you still want to continue. Type N to exit.
Input :4 / 2
Server replied:  2.0
Type the expression if you still want to continue. Type N to exit.
Input :
```

SERVER:

```
Connected to client:  ('127.0.0.1', 40454)
Connection closed by client.
Connected to client:  ('127.0.0.1', 40456)
Client send:   3 + 4
Sending:  7
Client send:   2
Sending:  Incoorect Syntax.
Client send:  2
Sending:  Incoorect Syntax.
Client send:  2 * 6
Sending:  12
Client send:  4 / 2
Sending:  2.0
```

CASE – II

In this case, a second client attempts to connect to the server when the server is already connected to a client. The server accepts the request and processes the second server's messages by forming a separate thread.

CLIENT:

```
uneednotknow@uneednotknow-VirtualBox:~/tut3$ python3 client.py 127.0.0.1 5000
Connected to server.
Input : 4 + 4
Server replied:  8
Type the expression if you still want to continue. Type N to exit.
Input : 3 - 7
Server replied:  -4
Type the expression if you still want to continue. Type N to exit.
Input :
```

SERVER:

```
Connected to client:  ('127.0.0.1', 40544)
Connection closed by client.
Connected to client:  ('127.0.0.1', 40546)
Client send:   4 + 4
Sending:  8
Client send:   3 - 7
Sending:  -4
```

CASE - III
The client is asked if they want to continue or not. If not, they are prompted to enter N.
At the client side, the connection is broken.
At the server side, it is ready to accept new connection requests.

CLIENT:

```
Type the expression if you still want to continue. Type N to exit.
Input :N
Session Over.
uneednotknow@uneednotknow-VirtualBox:~/tut3$
```

SERVER:

```
Connection closed by client.
```

SERVER - 3:

server3.py : Your server program "server3.py" will be a single process server that uses the "select" method to handle multiple clients concurrently.

Running server 3 and then connecting the client to it.

CASE - I:
In this case, after the client has connected to the server, they type in basic arithmetic equations. The server receives them, solves them and sends back the answer.

CLIENT:

```
uneednotknow@uneednotknow-VirtualBox:~/tut3$ python3 client.py 127.0.0.1 5001
Connected to server.
Input : 3 + 4
Server replied:  7
Type the expression if you still want to continue. Type N to exit.
Input : 3 * 4
Server replied:  12
Type the expression if you still want to continue. Type N to exit.
Input : 3 - 4
Server replied:  -1
```

```
Type the expression if you still want to continue. Type N to exit.
Input : 3 / 4
Server replied:  0.75
Type the expression if you still want to continue. Type N to exit.
Input : 3
Server replied:  Incorrect syntax.
Type the expression if you still want to continue. Type N to exit.
Input :3 ^ 4
Server replied:  Invalid operator.
Type the expression if you still want to continue. Type N to exit.
Input :3-4
Server replied:  Incorrect syntax.
Type the expression if you still want to continue. Type N to exit.
Input :
```

CASE – II:
Connecting multiple clients with the server.
The server accepts multiple requests and uses a select method to handle them all.

CLIENT II:

```
uneednotknow@uneednotknow-VirtualBox:~/tut3$ python3 client.py 127.0.0.1 5001
Connected to server.
Input : 6 - 4
Server replied:  2
Type the expression if you still want to continue. Type N to exit.
Input :
```

CLIENT III:

```
uneednotknow@uneednotknow-VirtualBox:~/tut3$ python3 client.py 127.0.0.1 5001
Connected to server.
Input : 8 + 7
Server replied:  15
Type the expression if you still want to continue. Type N to exit.
Input :8 * 4
Server replied:  32
Type the expression if you still want to continue. Type N to exit.
Input :8 / 4
Server replied:  2.0
Type the expression if you still want to continue. Type N to exit.
Input :8 - 4
Server replied:  4
Type the expression if you still want to continue. Type N to exit.
Input :
```

SERVER:

```
uneednotknow@uneednotknow-VirtualBox:~/tut3$ python3 server3.py 127.0.0.1 5001
Socket binded to 5001.
Socket is listening.
Connected to client:  ('127.0.0.1', 56008)
Connection closed with server by client:  ('127.0.0.1', 56008)
Connected to client:  ('127.0.0.1', 56010)
Message received from client:  ('127.0.0.1', 56010)
Client sent:   3 + 4
Sending:  7
Result sent to client ('127.0.0.1', 56010)
Message received from client:  ('127.0.0.1', 56010)
Client sent:   3 * 4
Sending:  12
Result sent to client ('127.0.0.1', 56010)
Message received from client:  ('127.0.0.1', 56010)
Client sent:   3 - 4
Sending:  -1
Result sent to client ('127.0.0.1', 56010)
Message received from client:  ('127.0.0.1', 56010)
Client sent:   3 / 4
Sending:  0.75
Result sent to client ('127.0.0.1', 56010)
Message received from client:  ('127.0.0.1', 56010)
Client sent:   3
Sending:  Incorrect syntax.
Result sent to client ('127.0.0.1', 56010)
Message received from client:  ('127.0.0.1', 56010)
Client sent:  3 ^ 4
Sending:  Invalid operator.
```

```
Result sent to client ('127.0.0.1', 56010)
Message received from client:  ('127.0.0.1', 56010)
Client sent:  3-4
Sending:  Incorrect syntax.
Result sent to client ('127.0.0.1', 56010)
Connected to client:  ('127.0.0.1', 56140)
Connection closed with server by client:  ('127.0.0.1', 56140)
Connected to client:  ('127.0.0.1', 56142)
Message received from client:  ('127.0.0.1', 56142)
Client sent:   8 + 7
Sending:  15
Result sent to client ('127.0.0.1', 56142)
Message received from client:  ('127.0.0.1', 56142)
Client sent:  8 * 4
Sending:  32
Result sent to client ('127.0.0.1', 56142)
Message received from client:  ('127.0.0.1', 56142)
Client sent:  8 / 4
Sending:  2.0
Result sent to client ('127.0.0.1', 56142)
Message received from client:  ('127.0.0.1', 56142)
Client sent:  8 - 4
Sending:  4
Result sent to client ('127.0.0.1', 56142)
Connected to client:  ('127.0.0.1', 56204)
Connection closed with server by client:  ('127.0.0.1', 56204)
Connected to client:  ('127.0.0.1', 56206)
Message received from client:  ('127.0.0.1', 56206)
Client sent:   6 - 4
Sending:  2
Result sent to client ('127.0.0.1', 56206)
```

CASE – III:
The client is asked if they want to continue or not. If not, they are prompted to enter N.
At the client side, the connection is broken.
Closing Clients II and III.

CLIENT II:

```
Type the expression if you still want to continue. Type N to exit.
Input :N
Session Over.
uneednotknow@uneednotknow-VirtualBox:~/tut3$
```

CLIENT III:

```
Type the expression if you still want to continue. Type N to exit.
Input :N
Session Over.
uneednotknow@uneednotknow-VirtualBox:~/tut3$
```

SERVER:

```
Connection closed with server by client:  ('127.0.0.1', 56142)
Connection closed with server by client:  ('127.0.0.1', 56206)
```

CASE –IV:
A new client attempts connection after the previous two clients have disconnected, and the server is still connected to the first server. The connection request is accepted by the server.

CLIENT IV:

```
uneednotknow@uneednotknow-VirtualBox:~/tut3$ python3 client.py 127.0.0.1 5001
Connected to server.
Input : 4 + 4
Server replied:  8
Type the expression if you still want to continue. Type N to exit.
Input :
```

SERVER:

```
Connected to client:  ('127.0.0.1', 56606)
Connection closed with server by client:  ('127.0.0.1', 56606)
Connected to client:  ('127.0.0.1', 56608)
Message received from client:  ('127.0.0.1', 56608)
Client sent:   4 + 4
Sending:  8
Result sent to client ('127.0.0.1', 56608)
```

CASE – V:
In this case, we try to connect to the port already in use.
The connection fails, and a error message is displayed:

```
uneednotknow@uneednotknow-VirtualBox:~/tut3$ python3 server3.py 127.0.0.1 5001
Port is engaged right now:  [Errno 98] Address already in use
uneednotknow@uneednotknow-VirtualBox:~/tut3$
```

SERVER – 4:

server4.py : Your server program "server4.py" will be an echo server (that replies the same message to the client that was received from the same client); it will be a single process server that uses the "select" method to handle multiple clients concurrently.

Running server 4 and then connecting the client to it.

CASE – I:
In this case, after the client has connected to the server, they type in basic arithmetic equations. The server receives them,and sends back the same message, irrespective of what the message is.

CLIENT:

```
uneednotknow@uneednotknow-VirtualBox:~/tut3$ python3 client.py 127.0.0.1 5003
Connected to server.
Input : 3 + 4
Server replied:   3 + 4
Type the expression if you still want to continue. Type N to exit.
Input : 5 * 4
Server replied:   5 * 4
Type the expression if you still want to continue. Type N to exit.
Input : 6 / 3
Server replied:   6 / 3
Type the expression if you still want to continue. Type N to exit.
Input :9 - 9
Server replied:  9 - 9
Type the expression if you still want to continue. Type N to exit.
Input :
```

SERVER:

```
uneednotknow@uneednotknow-VirtualBox:~/tut3$ python3 server4.py 127.0.0.1 5003
Socket binded to 5003.
Socket is listening.
Connected to client:  ('127.0.0.1', 42880)
Connection closed with server by client:  ('127.0.0.1', 42880)
Connected to client:  ('127.0.0.1', 42882)
Message received from client:  ('127.0.0.1', 42882)
Client sent:   3 + 4
Sending:   3 + 4
Result sent to client ('127.0.0.1', 42882)
Message received from client:  ('127.0.0.1', 42882)
Client sent:   5 * 4
Sending:   5 * 4
Result sent to client ('127.0.0.1', 42882)
```

```
Message received from client:  ('127.0.0.1', 42882)
Client sent:   6 / 3
Sending:   6 / 3
Result sent to client ('127.0.0.1', 42882)
Message received from client:  ('127.0.0.1', 42882)
Client sent:  9 - 9
Sending:  9 - 9
Result sent to client ('127.0.0.1', 42882)
```

CASE – II:
In this case, the client sends some wrong inputs. The server distinguish thee wrong inputs and sends back an appropriate error message:

CLIENT:

```
Type the expression if you still want to continue. Type N to exit.
Input : 6
Server replied:   6
Type the expression if you still want to continue. Type N to exit.
Input : 6 & 4
Server replied:   6 & 4
Type the expression if you still want to continue. Type N to exit.
Input :6-4
Server replied:  6-4
Type the expression if you still want to continue. Type N to exit.
Input :
```

SERVER:

```
Message received from client:  ('127.0.0.1', 42882)
Client sent:   6
Sending:   6
Result sent to client ('127.0.0.1', 42882)
Message received from client:  ('127.0.0.1', 42882)
Client sent:   6 & 4
Sending:   6 & 4
Result sent to client ('127.0.0.1', 42882)
Message received from client:  ('127.0.0.1', 42882)
Client sent:  6-4
Sending:  6-4
Result sent to client ('127.0.0.1', 42882)
```

CASE – III:
Connecting multiple clients with the server. The server accepts multiple requests and uses a select method to handle them all.

CLIENT II:

```
uneednotknow@uneednotknow-VirtualBox:~/tut3$ python3 client.py 127.0.0.1 5003
Connected to server.
Input : server 4
Server replied:   server 4
Type the expression if you still want to continue. Type N to exit.
Input : 4 - 2
Server replied:   4 - 2
Type the expression if you still want to continue. Type N to exit.
Input :S
```

CLIENT III:

```
uneednotknow@uneednotknow-VirtualBox:~/tut3$ python3 client.py 127.0.0.1 5003
Connected to server.
Input : hello
Server replied:   hello
Type the expression if you still want to continue. Type N to exit.
Input :
```

SERVER:

```
Connected to client:  ('127.0.0.1', 43218)
Connection closed with server by client:  ('127.0.0.1', 43218)
Connected to client:  ('127.0.0.1', 43220)
Message received from client:  ('127.0.0.1', 43220)
Client sent:   hello
Sending:   hello
Result sent to client ('127.0.0.1', 43220)
Connected to client:  ('127.0.0.1', 43258)
Connected to client:  ('127.0.0.1', 43260)
Connection closed with server by client:  ('127.0.0.1', 43258)
Message received from client:  ('127.0.0.1', 43260)
Client sent:   server 4
Sending:   server 4
Result sent to client ('127.0.0.1', 43260)
Message received from client:  ('127.0.0.1', 43260)
Client sent:   4 - 2
Sending:   4 - 2
Result sent to client ('127.0.0.1', 43260)
```

CASE – IV:

In this case, we try to connect to the port already in use.

The connection fails, and a error message is displayed:

```
uneednotknow@uneednotknow-VirtualBox:~/tut3$ python3 server3.py 127.0.0.1 5003
Port is engaged right now:  [Errno 98] Address already in use
uneednotknow@uneednotknow-VirtualBox:~/tut3$
```

CASE – V:

The client is asked if they want to continue or not. If not, they are prompted to enter N.

At the client side, the connection is broken.

Closing Clients II and III.

CLIENT:

```
Type the expression if you still want to continue. Type N to exit.
Input :N
Session Over.
uneednotknow@uneednotknow-VirtualBox:~/tut3$
```

SERVER:

```
Connection closed with server by client:  ('127.0.0.1', 42882)
```
