

CS359: Computer Networks Lab

Tut03 - Assignment 3

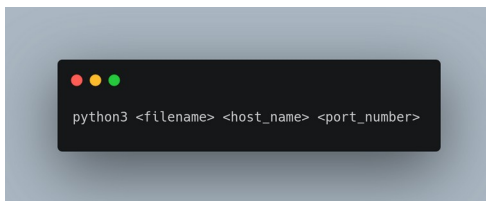
Name: Kartik Mouli

Roll No.: 2001CS35

Note: Ubuntu 20.04 OS and Python 3.10.6 used for this Assignment.

Commands to run code:

To run the server, open a terminal or command prompt and navigate to the directory . Then, run the following command:



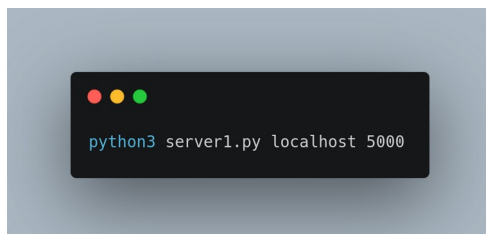
```
python3 <filename> <host_name> <port_number>
```

filename : server1.py, server2.py, server3.py, server4.py, client.py

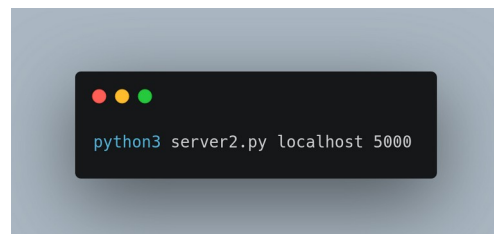
hostname: localhost i.e 127.0.0.1

portnumber: port to be used

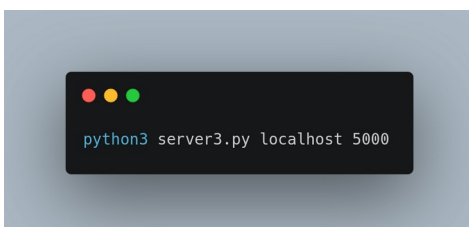
Replace <host_name> and <port_number> with the IP address and port number you want the server to listen on, respectively. For example, to run the server on localhost port 5000, you can use:



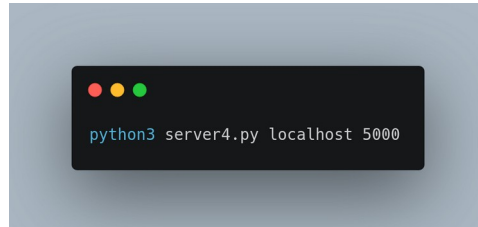
```
python3 server1.py localhost 5000
```



```
python3 server2.py localhost 5000
```

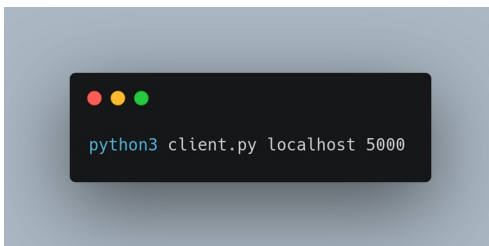


```
python3 server3.py localhost 5000
```



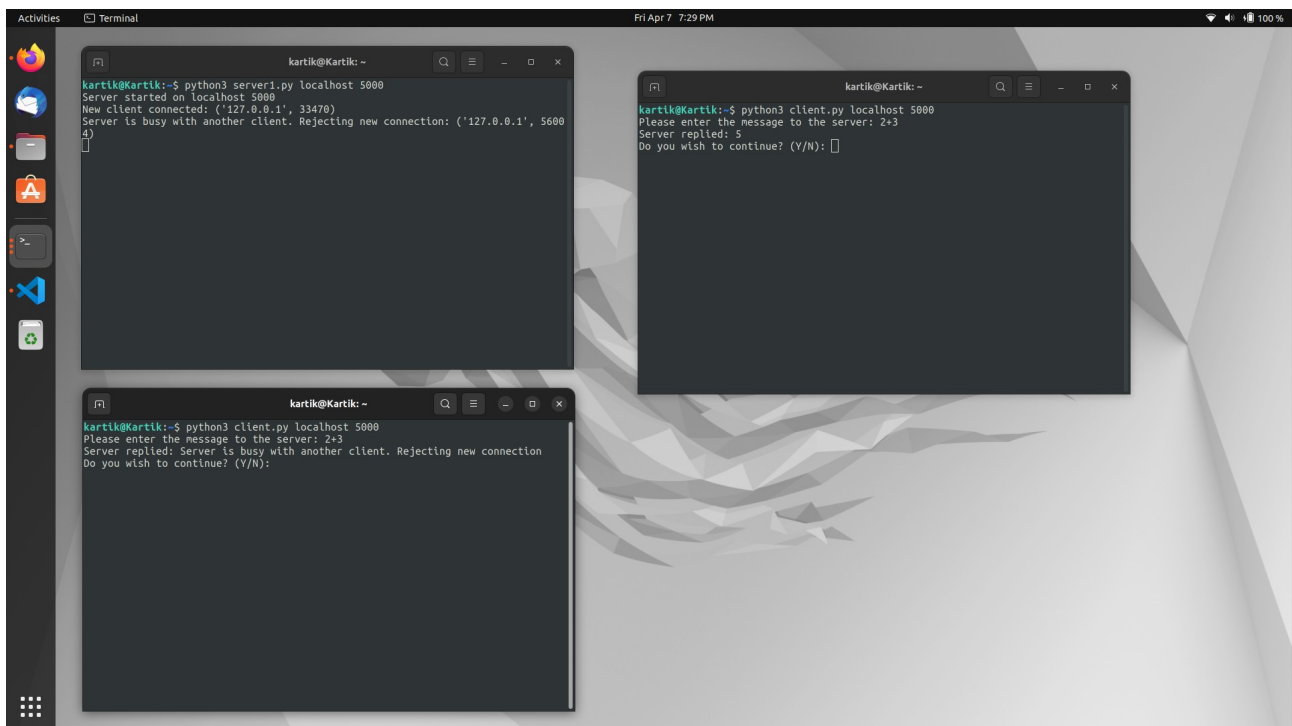
```
python3 server4.py localhost 5000
```

To run the client, open another terminal or command prompt. Then, run the following command:

A terminal window with a dark background and light blue text. The command `python3 client.py localhost 5000` is entered at the prompt.

```
python3 client.py localhost 5000
```

Output of Server 1:

Three terminal windows are shown. The top-left window shows the server starting and accepting a connection. The top-right window shows the client sending a message and receiving a reply. The bottom window shows the client sending a message and receiving a reply, followed by the server rejecting a new connection.

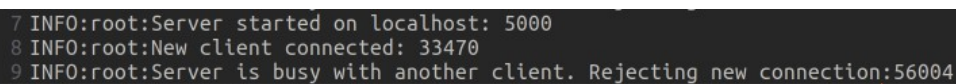
```
kartik@Kartik:~$ python3 server1.py localhost 5000
Server started on localhost 5000
New client connected: ('127.0.0.1', 33470)
Server is busy with another client. Rejecting new connection: ('127.0.0.1', 56004)

kartik@Kartik:~$ python3 client.py localhost 5000
Please enter the message to the server: 2+3
Server replied: 5
Do you wish to continue? (Y/N):

kartik@Kartik:~$ python3 client.py localhost 5000
Please enter the message to the server: 2+3
Server replied: Server is busy with another client. Rejecting new connection
Do you wish to continue? (Y/N):
```

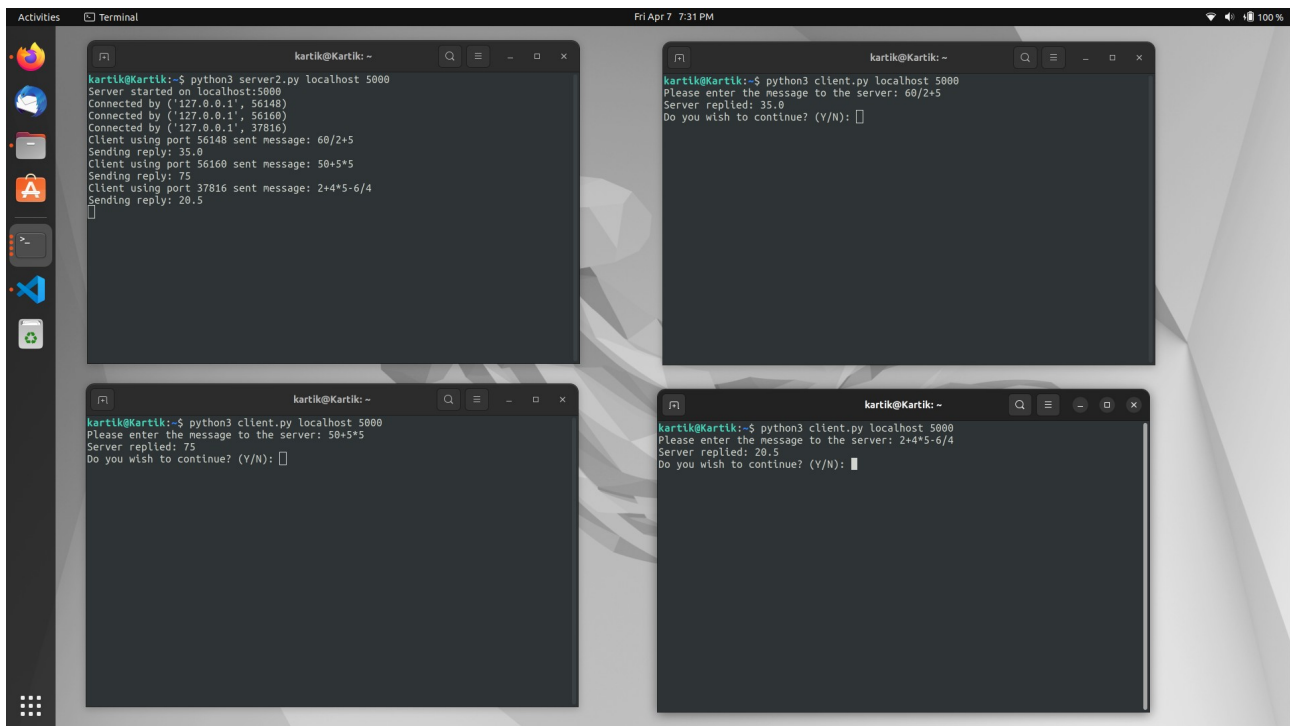
As we can see that if another client tries to connect to server, It will rejects the connection and shows an error message.

Logfile of server 1:

A log file with three lines of text. The first line shows the server starting on localhost: 5000. The second line shows a new client connected with IP 127.0.0.1 and port 33470. The third line shows the server rejecting a new connection from IP 127.0.0.1 and port 56004.

```
7 INFO:root:Server started on localhost: 5000
8 INFO:root:New client connected: 33470
9 INFO:root:Server is busy with another client. Rejecting new connection:56004
```

Output of Server 2:

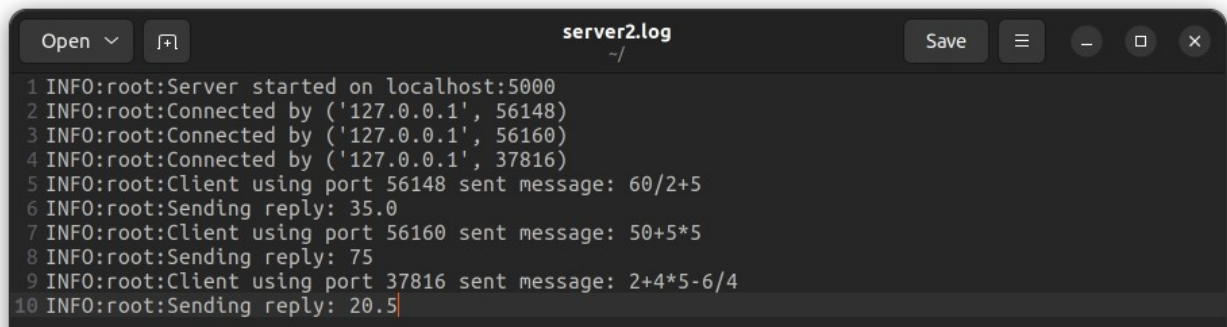


The image displays four terminal windows from a Linux desktop environment. The top-left window shows the server's output: it starts on localhost:5000, connects to three clients (56148, 56160, 37816), and processes their requests. The top-right window shows a client (56148) sending '60/2+5' and receiving '35.0'. The bottom-left window shows a client (56160) sending '50+5*5' and receiving '75'. The bottom-right window shows a client (37816) sending '2+4*5-6/4' and receiving '20.5'.

```
kartik@Kartik: ~  
$ python3 server2.py localhost 5000  
Server started on localhost:5000  
Connected by ('127.0.0.1', 56148)  
Connected by ('127.0.0.1', 56160)  
Connected by ('127.0.0.1', 37816)  
Client using port 56148 sent message: 60/2+5  
Sending reply: 35.0  
Client using port 56160 sent message: 50+5*5  
Sending reply: 75  
Client using port 37816 sent message: 2+4*5-6/4  
Sending reply: 20.5  
$  
  
kartik@Kartik: ~  
$ python3 client.py localhost 5000  
Please enter the message to the server: 60/2+5  
Server replied: 35.0  
Do you wish to continue? (Y/N):  
  
kartik@Kartik: ~  
$ python3 client.py localhost 5000  
Please enter the message to the server: 50+5*5  
Server replied: 75  
Do you wish to continue? (Y/N):  
  
kartik@Kartik: ~  
$ python3 client.py localhost 5000  
Please enter the message to the server: 2+4*5-6/4  
Server replied: 20.5  
Do you wish to continue? (Y/N):
```

Multithreaded server, as we can see that multiple clients can use the server.

Logfile of server 2:

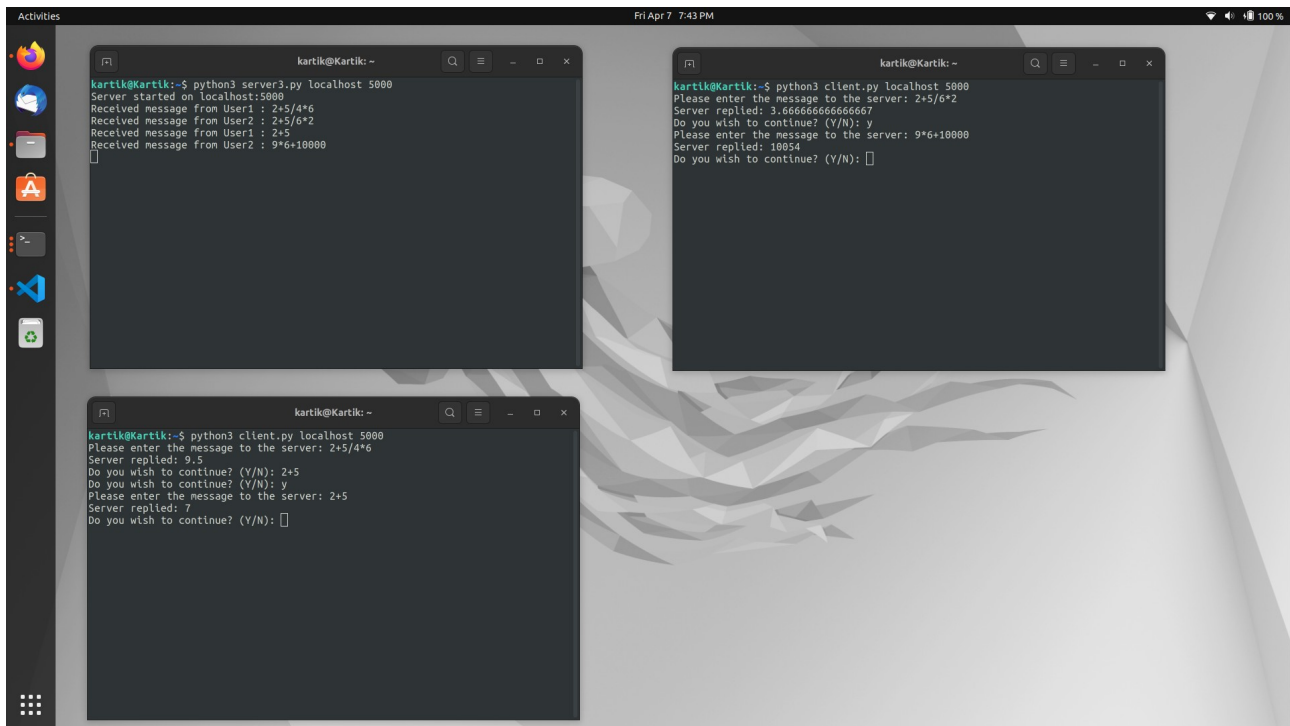


The image shows a log file window titled 'server2.log' with a list of 10 log entries. Each entry is prefixed with a line number and the text 'INFO:root:'. The entries correspond to the server's startup, client connections, message processing, and replies.

```
server2.log  
1 INFO:root:Server started on localhost:5000  
2 INFO:root:Connected by ('127.0.0.1', 56148)  
3 INFO:root:Connected by ('127.0.0.1', 56160)  
4 INFO:root:Connected by ('127.0.0.1', 37816)  
5 INFO:root:Client using port 56148 sent message: 60/2+5  
6 INFO:root:Sending reply: 35.0  
7 INFO:root:Client using port 56160 sent message: 50+5*5  
8 INFO:root:Sending reply: 75  
9 INFO:root:Client using port 37816 sent message: 2+4*5-6/4  
10 INFO:root:Sending reply: 20.5
```

It shows that client using port_number sends a request and the servers respond to that request.

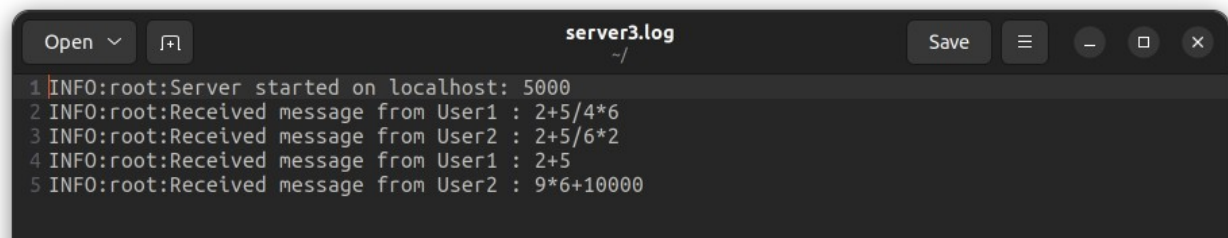
Output of Server 3:



```
kartik@Kartik: ~  
kartik@Kartik:~$ python3 server3.py localhost 5000  
Server started on localhost:5000  
Received message from User1 : 2+5/4*6  
Received message from User2 : 2+5/6*2  
Received message from User1 : 2+5  
Received message from User2 : 9*6+10000  
[ ]  
  
kartik@Kartik:~$ python3 client.py localhost 5000  
Please enter the message to the server: 2+5/6*2  
Server replied: 3.6666666666666667  
Do you wish to continue? (Y/N): y  
Please enter the message to the server: 9*6+10000  
Server replied: 10054  
Do you wish to continue? (Y/N): [ ]  
  
kartik@Kartik:~$ python3 client.py localhost 5000  
Please enter the message to the server: 2+5/4*6  
Server replied: 9.5  
Do you wish to continue? (Y/N): 2+5  
Do you wish to continue? (Y/N): y  
Please enter the message to the server: 2+5  
Server replied: 7  
Do you wish to continue? (Y/N): [ ]
```

single process server that uses the “select” method to handle multiple clients concurrently.

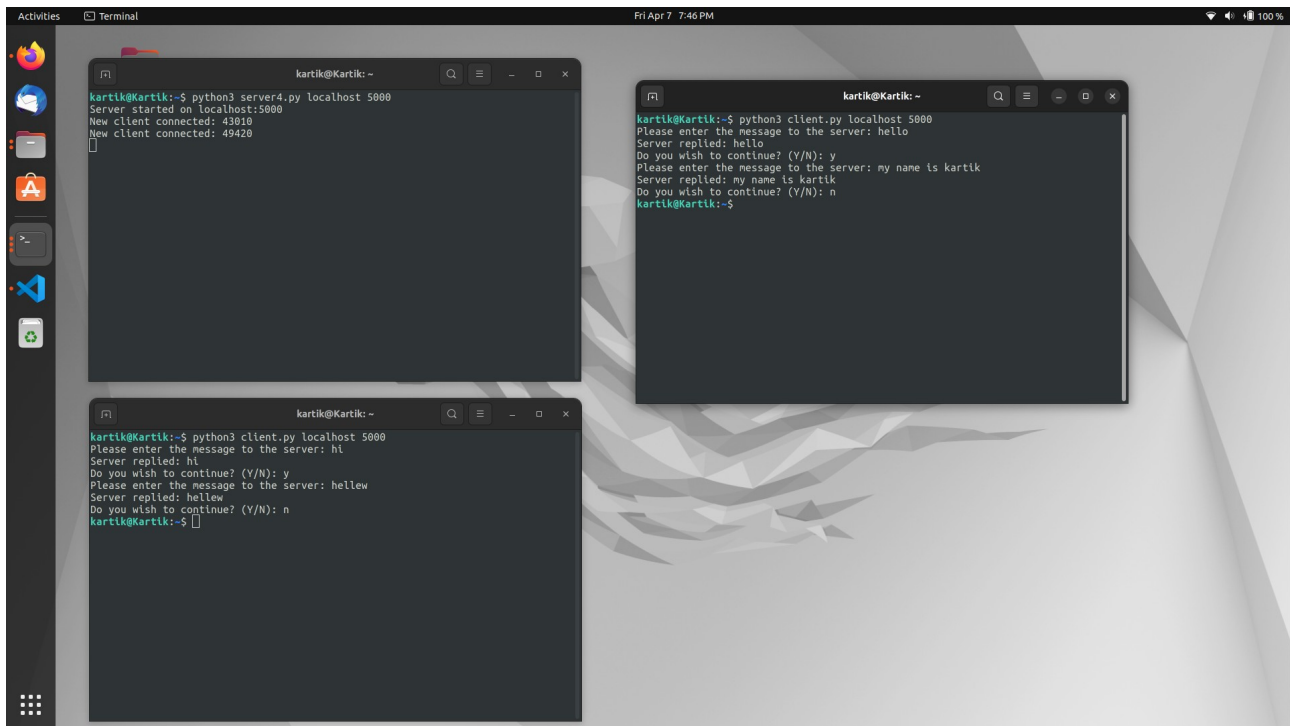
Logfile of Server 3:



```
server3.log  
~/  
1 INFO:root:Server started on localhost: 5000  
2 INFO:root:Received message from User1 : 2+5/4*6  
3 INFO:root:Received message from User2 : 2+5/6*2  
4 INFO:root:Received message from User1 : 2+5  
5 INFO:root:Received message from User2 : 9*6+10000
```

It shows that which user sends which request.

Output of Server 4:



The image shows three terminal windows on a Linux desktop. The top-left window shows the server output: `python3 server4.py localhost 5000` results in 'Server started on localhost:5000', followed by two client connections at ports 43010 and 49420. The top-right window shows the client output for the first connection: `python3 client.py localhost 5000` prompts for a message ('hello'), which is echoed back, and asks if the user wants to continue (yes). The bottom window shows the client output for the second connection: prompts for a message ('helloworld'), which is echoed back, and asks if the user wants to continue (no).

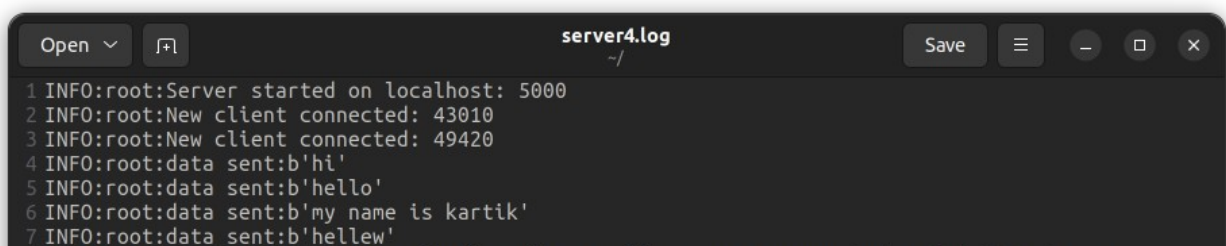
```
kartik@Kartik:~$ python3 server4.py localhost 5000
Server started on localhost:5000
New client connected: 43010
New client connected: 49420

kartik@Kartik:~$ python3 client.py localhost 5000
Please enter the message to the server: hello
Server replied: hello
Do you wish to continue? (Y/N): y
Please enter the message to the server: my name is kartik
Server replied: my name is kartik
Do you wish to continue? (Y/N): n
kartik@Kartik:~$

kartik@Kartik:~$ python3 client.py localhost 5000
Please enter the message to the server: hi
Server replied: hi
Do you wish to continue? (Y/N): y
Please enter the message to the server: helloworld
Server replied: helloworld
Do you wish to continue? (Y/N): n
kartik@Kartik:~$
```

This is a echo server which replies the same message to the client that was received from the same client.

Logfile of Server 4:



The image shows a text editor window titled 'server4.log' with the following log entries:

```
1 INFO:root:Server started on localhost: 5000
2 INFO:root:New client connected: 43010
3 INFO:root:New client connected: 49420
4 INFO:root:data sent:b'hi'
5 INFO:root:data sent:b'hello'
6 INFO:root:data sent:b'my name is kartik'
7 INFO:root:data sent:b'helloworld'
```

This file consists of clients port number and request made by client.

Conclusion :

The codes are running perfectly.

Additional feature implemented are logging which create a logfile of each server and if command line arguments are incorrect it shows which arguments to be used.

Link of Video Demo:

[for video click here](#)