Computer Network Lab
CS359

Tanishq Malu Lab:3 1901CS63

Lab Topic : Socket Programming in Python

Question Number : FileName

Question1 : server.py
Question2 : multiclient.py

Additional Files

create_config.py : It is used to create config file in json format, which is later used in configuration multiclient server.

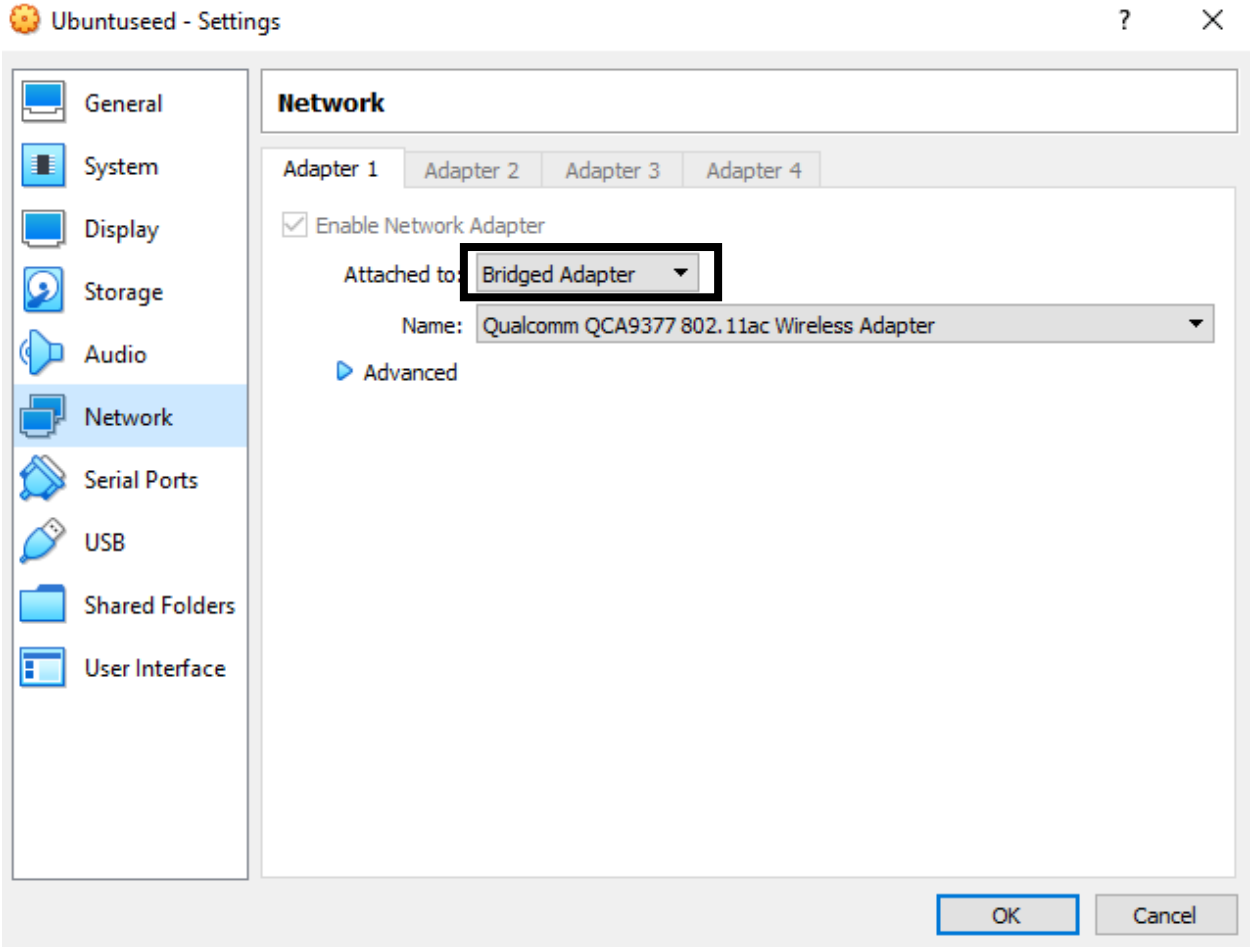
sample.json : Config file created by the create_config.py

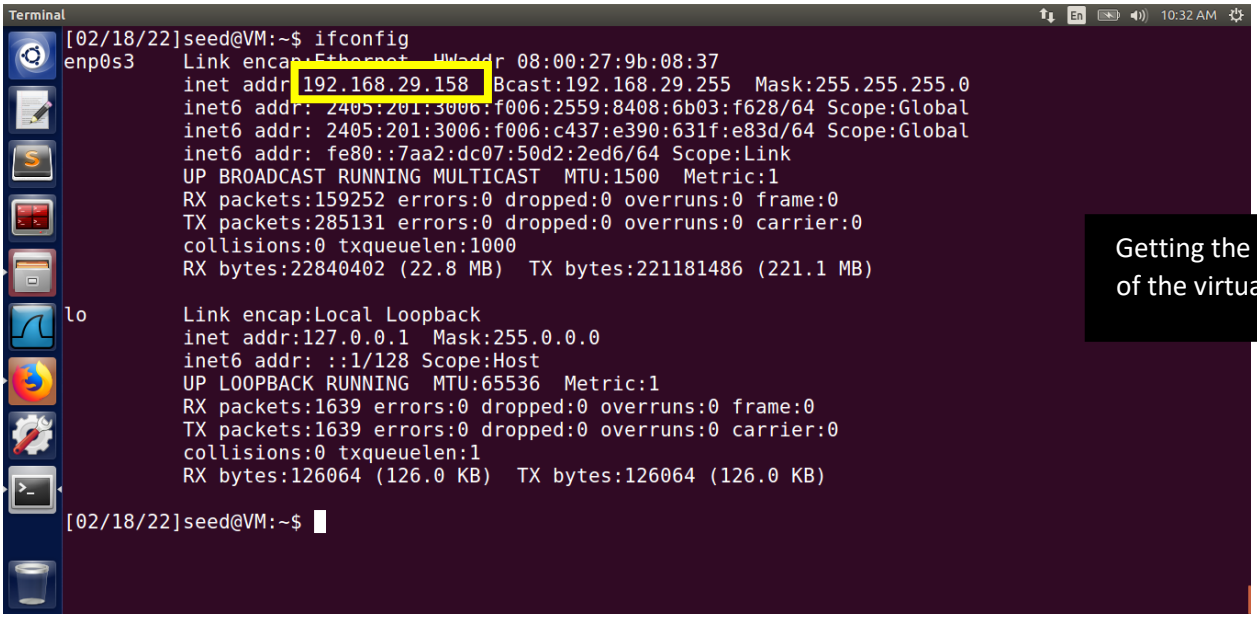
Socket.txt : Files which client can request
hello.html : Files which client can request

(Please turn over for detailed report)

Our basic aim is to create a server-client model using socket programming in python. For this purpose, I'll be running my server code on my ubuntu virtual machine and host my server on it. Simultaneously, I'll try to send requests as a client from my windows host.

So, First let us find the ip address on which we will host the server, i.e. IP address of our virtual machine. For this purpose, minor settings should be made in your VM so that your original host computer consider it as an another network. Network settings of VM should be set as bridged adapter to work this out.

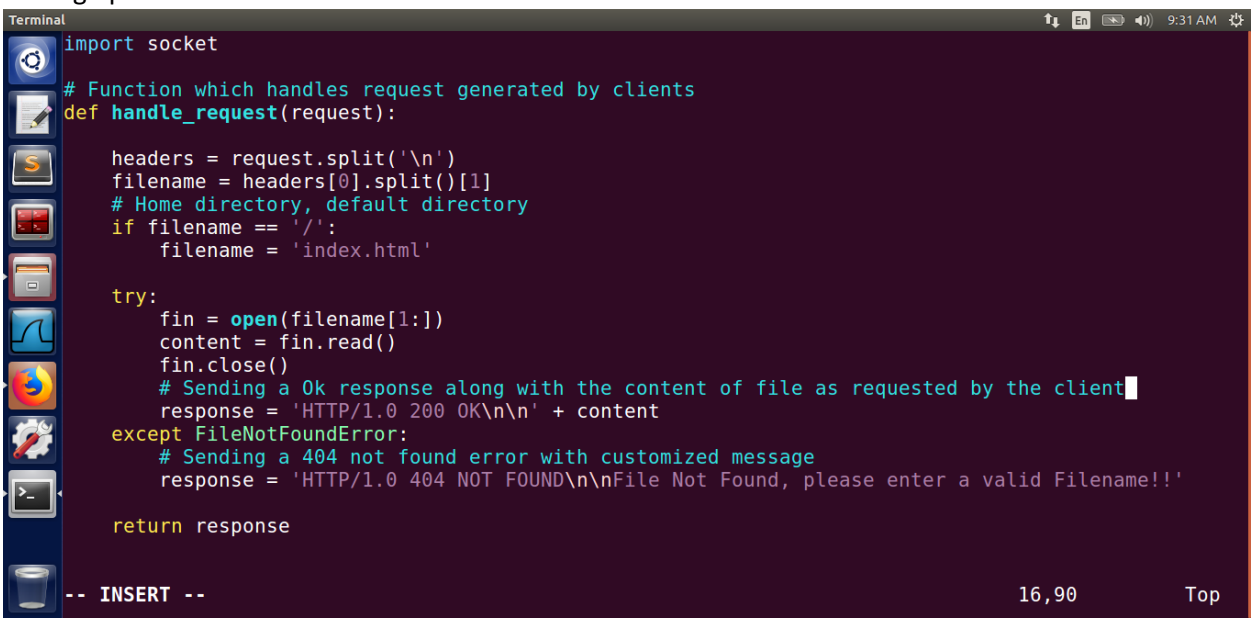




Getting the IP address of the virtual machine

Question-1:

Setting up the Server code:



```
# Define socket host and port
SERVER_HOST = "192.168.29.158" # Ip address of the host
SERVER_PORT = 8888             # Port used for communication between server and client

# Create socket
server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server_socket.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
server_socket.bind((SERVER_HOST, SERVER_PORT))
server_socket.listen(1)
print(server_socket,' Listening on port %s ...' % SERVER_PORT)

while True:
    # Wait for client connections
    client_connection, client_address = server_socket.accept()
    # Get the client request
    request = client_connection.recv(1024).decode()
    print("Client is connected with Ip:",client_address)
    # Return an HTTP response
    response = handle_request(request)
    client_connection.sendall(response.encode())
    print("Client has disconnected with Ip:",client_address)
    # Close connection
    client_connection.close()

-- INSERT --                                                    48,1          92%
```

Using port number 8888. Usually port number greater than 1024 are not reserved and can be used for our experiment

Running the server:

```
[02/18/22]seed@VM:~$ vim server.py
[02/18/22]seed@VM:~$ python3 server.py
Listening on port 8888 ...
```

```
[02/18/22]seed@VM:~$ vim server.py
[02/18/22]seed@VM:~$ python3 server.py
Listening on port 8888 ...
Client is connected with Ip: ('192.168.29.138', 64208)
Client has disconnected with Ip: ('192.168.29.138', 64208)
```

Page Title

Not secure | http://192.168.29.158:8888/hello.html

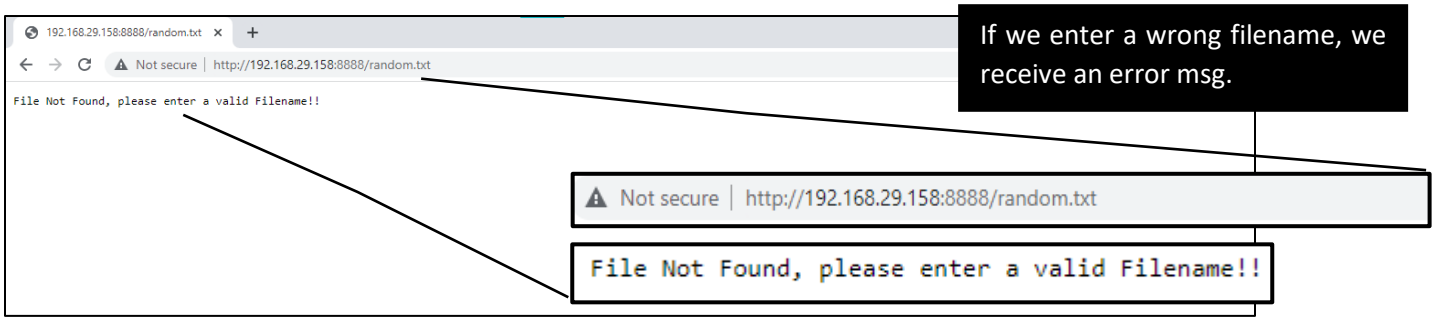
Hello world !!!!

Socket programming lab

Tanishq Malu - 1901CS63

If we enter a valid filename, the file gets opened.

Not secure | http://192.168.29.158:8888/hello.html



Question-2:

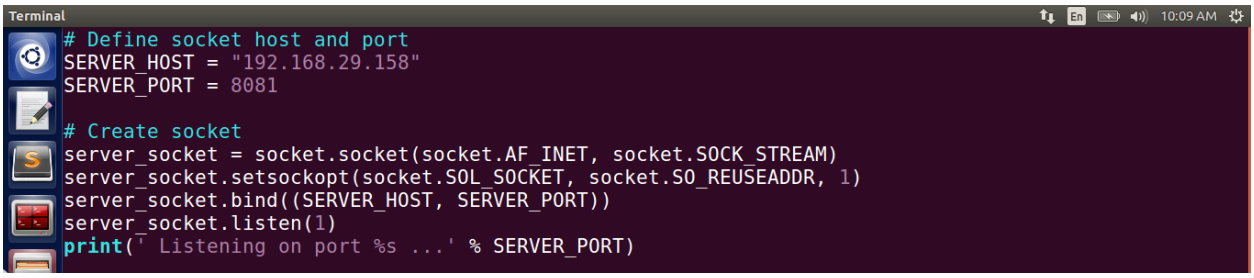
```
Terminal                                                    9:55 AM
import socket
import threading
import _thread
import json
import time

# Reading config file
with open('sample.json', 'r') as openfile:
    # Reading from json file
    config = json.load(openfile)

# Default values, accessed from the config file

# Default filename
default_filename = config["defaultFile"]
# Default maximum concurrent connections allowed
max_concurrent_con = config["max_concurrent_thread"]
# Default IPs which are blocked
block_ip = config["blockIp"]

# Global variable to keep a count
# current number of concurrent connection
current_concurrent_con = 0

-- INSERT --                                          22,42         Top
```

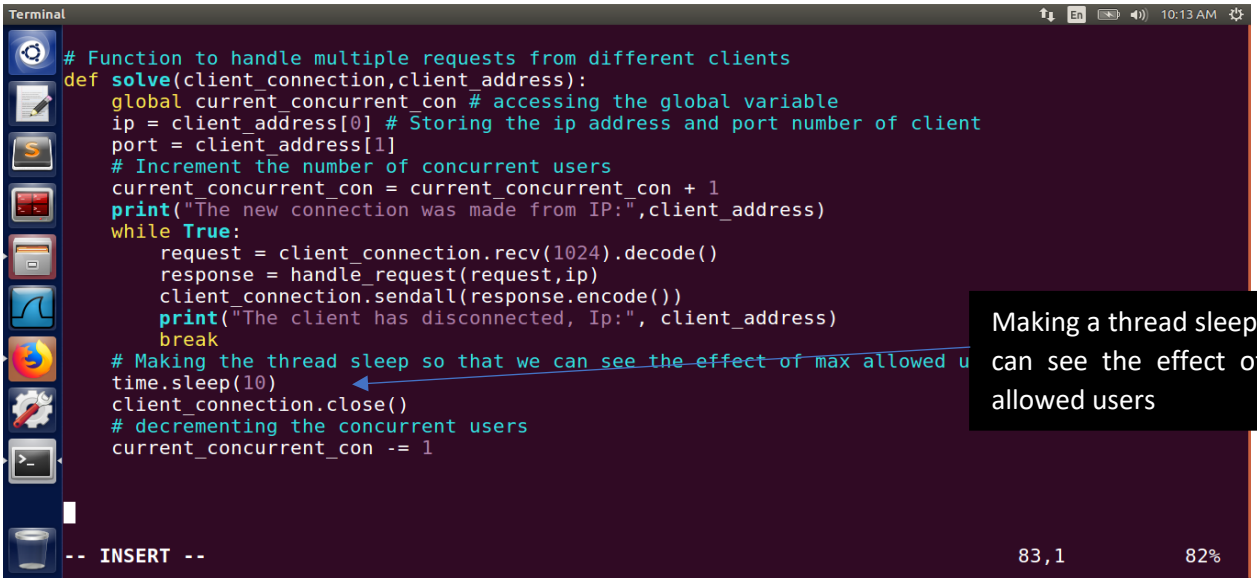
Setting up default values from config file.

```
Terminal                                                    10:03 AM
#Function to handle requests generated by clients
def handle_request(request,ip):
    global current_concurrent_con # accessing the global variable
    headers = request.split('\n')
    filename = headers[0].split()[1] # grabbing the filename from the header

    if filename == '/':
        filename = default_filename # redirecting to home directory/default

    try:
        fin = open(filename[1:])
        content = fin.read() # reading the contents of the file
        fin.close()
        # Generating an OK response for client
        response = 'HTTP/1.0 200 OK\n\n' + content
    except FileNotFoundError:
        #generating a 404 error with a cutomized msg
        response = 'HTTP/1.0 404 NOT FOUND\n\nFile Not Found!!!'
    # Blocking the response, if ip is in blocked ip list
    if block_ip.count(ip)>0:
        response = 'HTTP/1.0 404 NOT FOUND\n\n Your Ip is blocked!!!'

    if current_concurrent_con > max_concurrent_con:
        response = 'HTTP/1.0 404 NOT FOUND\n\n Server Overload, User limit exceeded !!!'
-- INSERT --                                          48,88         34%
```

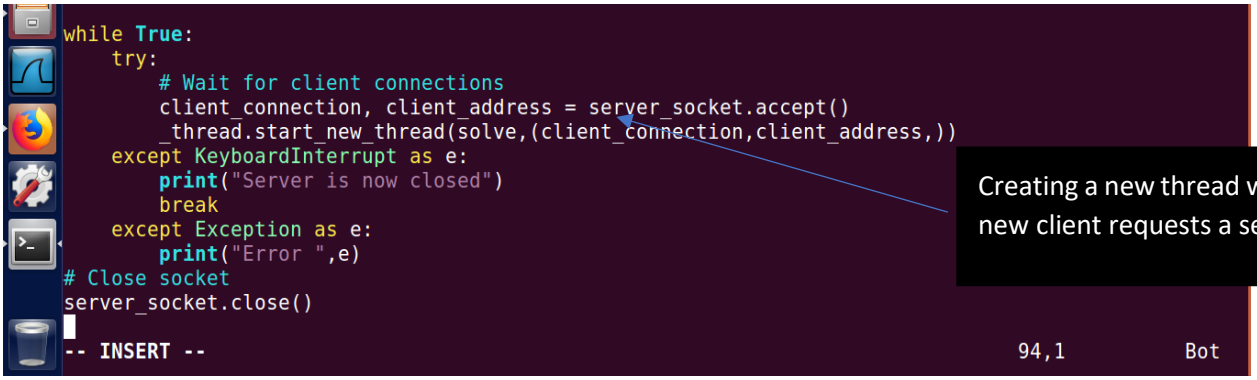
Checking current IP with blocked IP

Checking number of concurrent clients requesting to server

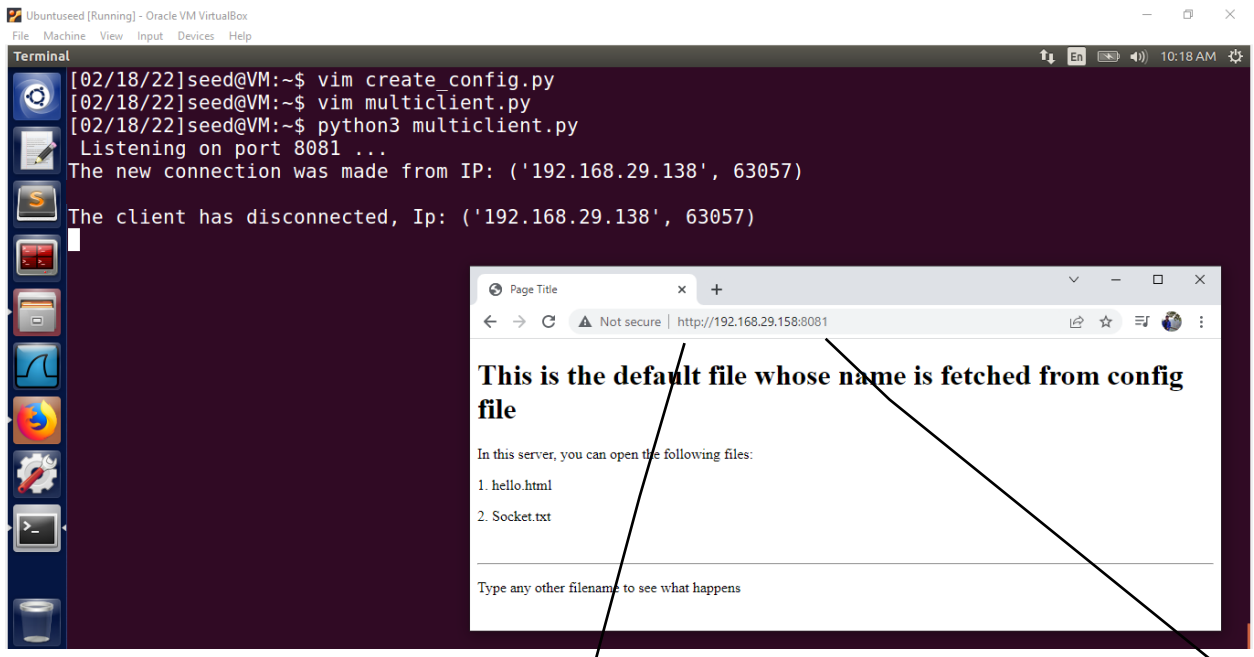

Terminal
10:09 AM
# Define socket host and port
SERVER_HOST = "192.168.29.158"
SERVER_PORT = 8081
# Create socket
server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server_socket.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
server_socket.bind((SERVER_HOST, SERVER_PORT))
server_socket.listen(1)
print(' Listening on port %s ...' % SERVER_PORT)


Terminal
10:13 AM
# Function to handle multiple requests from different clients
def solve(client_connection,client_address):
global current_concurrent_con # accessing the global variable
ip = client_address[0] # Storing the ip address and port number of client
port = client_address[1]
# Increment the number of concurrent users
current_concurrent_con = current_concurrent_con + 1
print("The new connection was made from IP:",client_address)
while True:
request = client_connection.recv(1024).decode()
response = handle_request(request,ip)
client_connection.sendall(response.encode())
print("The client has disconnected, Ip:", client_address)
break
# Making the thread sleep so that we can see the effect of max allowed u
time.sleep(10)
client_connection.close()
# decrementing the concurrent users
current_concurrent_con -= 1
-- INSERT --
83,1
82%
Making a thread sleep, so that we can see the effect of maximum allowed users

t

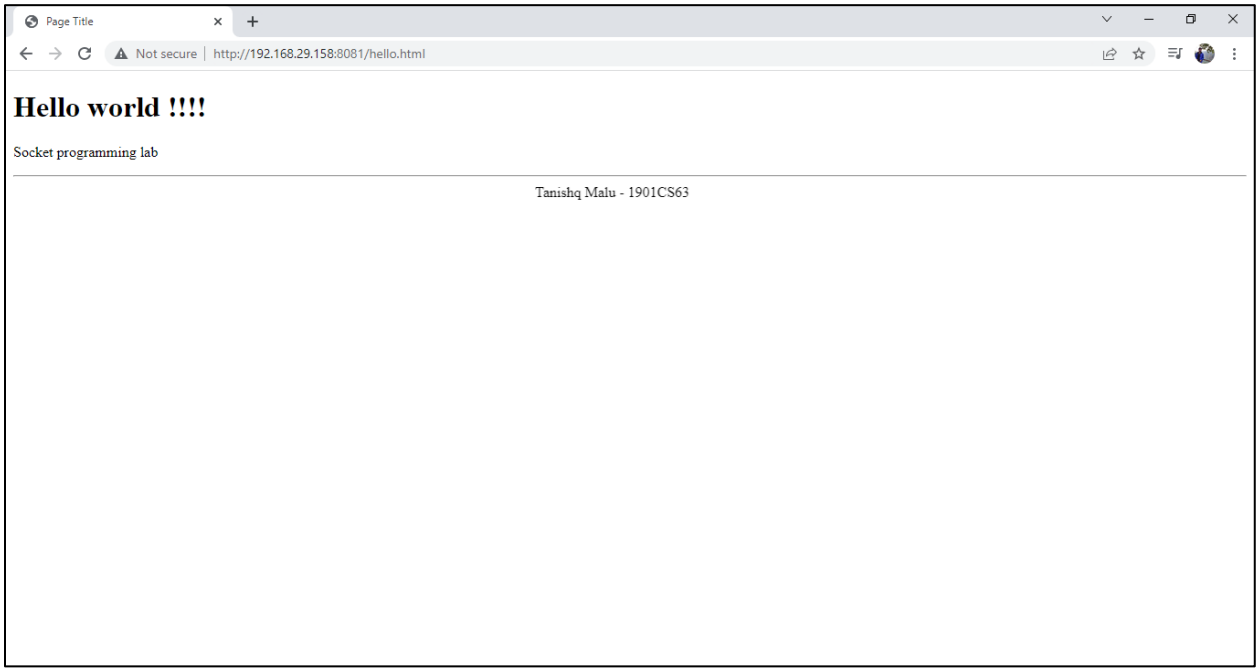

while True:
try:
# Wait for client connections
client_connection, client_address = server_socket.accept()
_thread.start_new_thread(solve,(client_connection,client_address,))
except KeyboardInterrupt as e:
print("Server is now closed")
break
except Exception as e:
print("Error ",e)
# Close socket
server_socket.close()
-- INSERT --
94,1
Bot
Creating a new thread whenever a new client requests a server

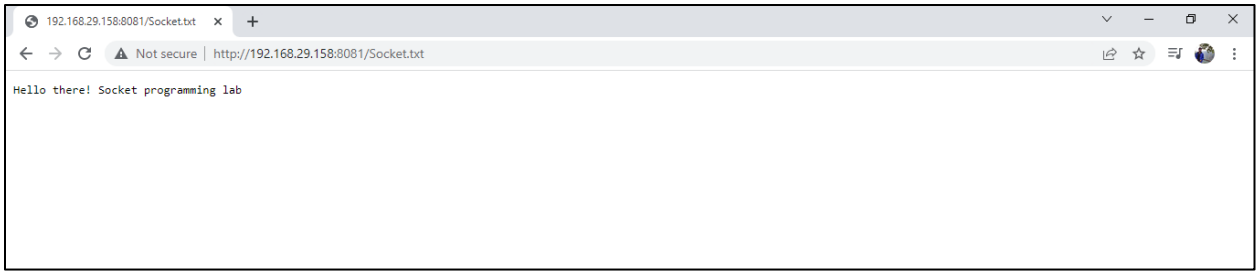
Running the server code:



Although a file name was not given but the server directed it to a default file mentioned in the config file

Opening different files in multiple tabs(less than max_concurrent_con)

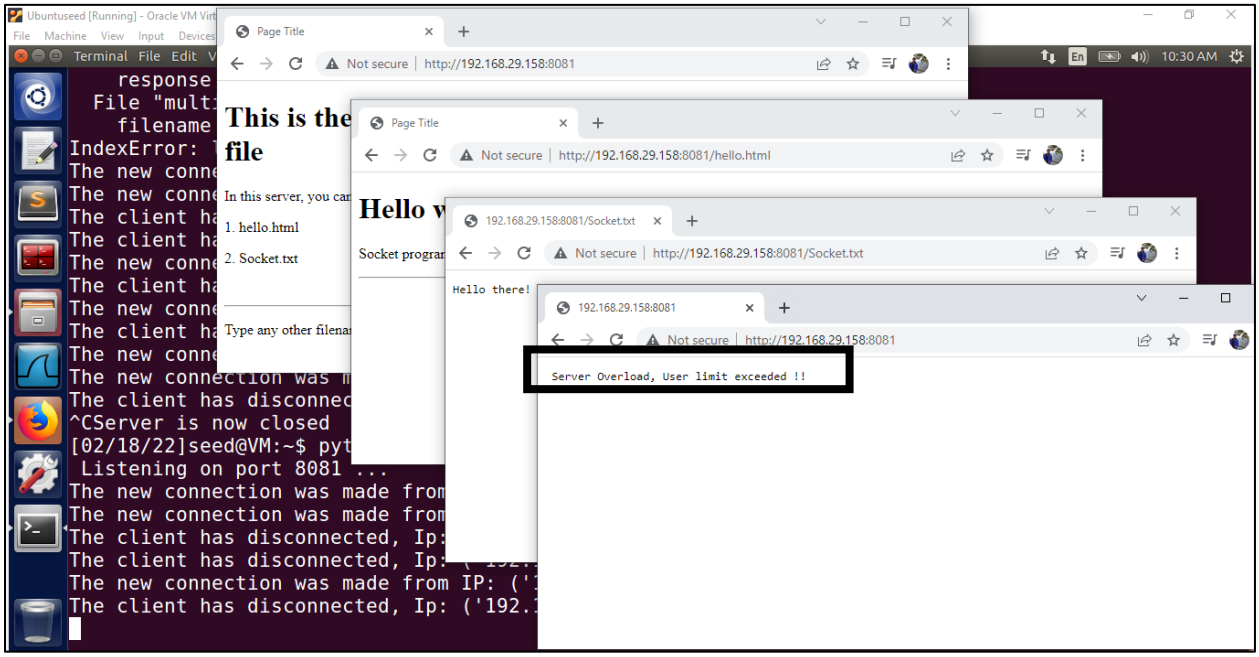




Max concurrent users allowed = 3 (in my config file)

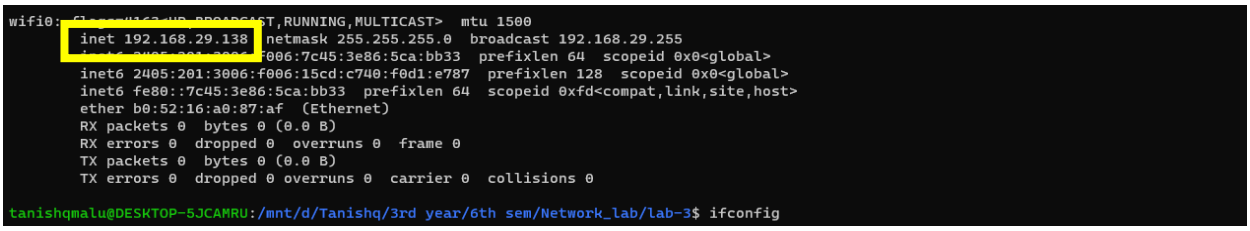
So, when I try to make 4 concurrent requests

```
The new connection was made from IP: ('192.168.29.138', 63343)
The new connection was made from IP: ('192.168.29.138', 63342)
The client has disconnected, Ip: ('192.168.29.138', 63342)
The client has disconnected, Ip: ('192.168.29.138', 63343)
The new connection was made from IP: ('192.168.29.138', 63346)
The client has disconnected, Ip: ('192.168.29.138', 63346)
The new connection was made from IP: ('192.168.29.138', 63348)
The client has disconnected, Ip: ('192.168.29.138', 63348)
```



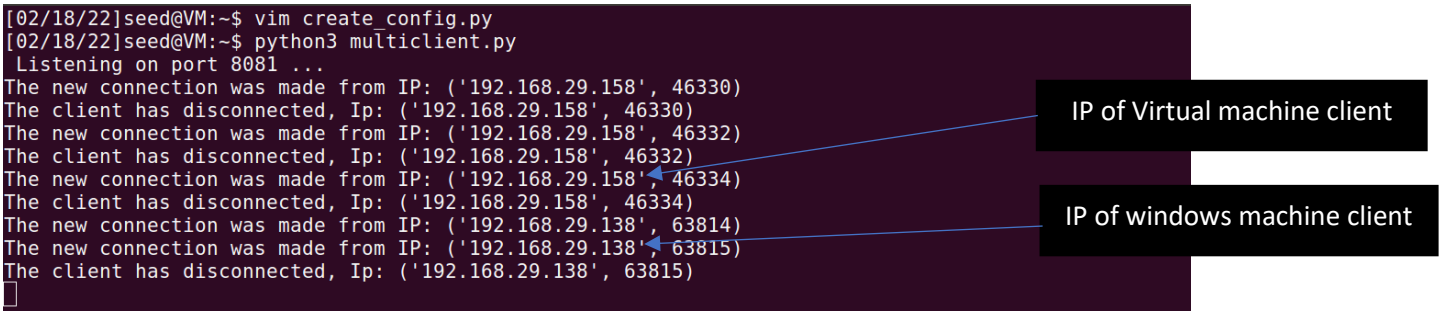
If I modify the config file and add my windows machine IP address in blocked IP, I will not be allowed to access the file.

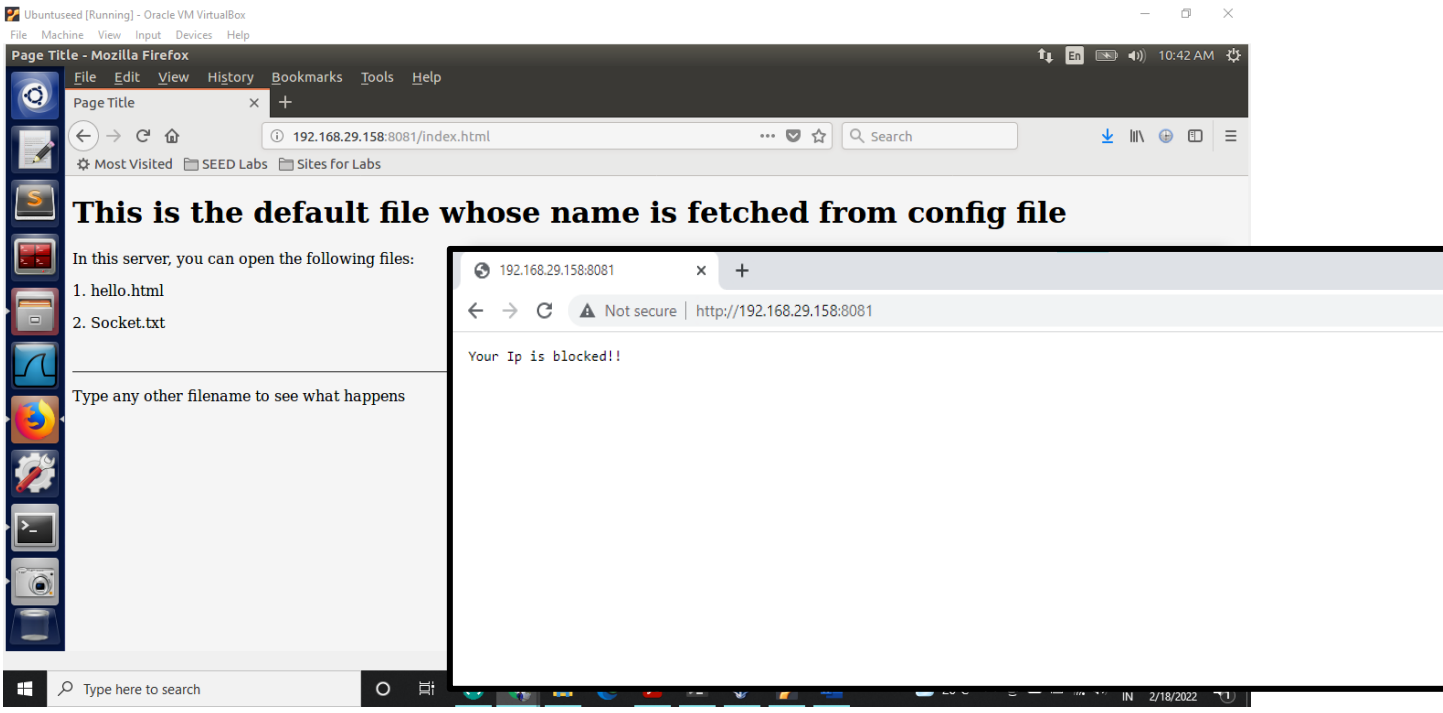
Getting my ip address of windows machine:



Creating the config json file:

Now on running the server again, I can't access the server from my windows machine but can accessit via virtual machine because its ip address is not mentioned in the blocked ip list.





----------End Of Assignment----------