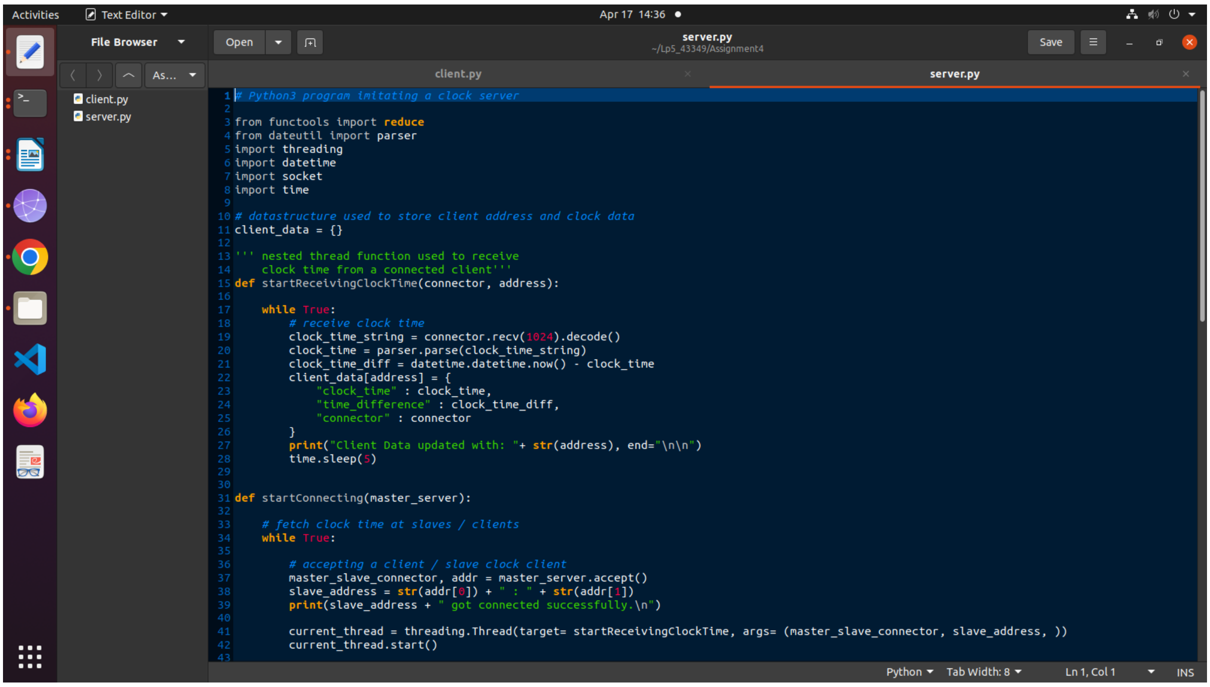
Roll.No-43349

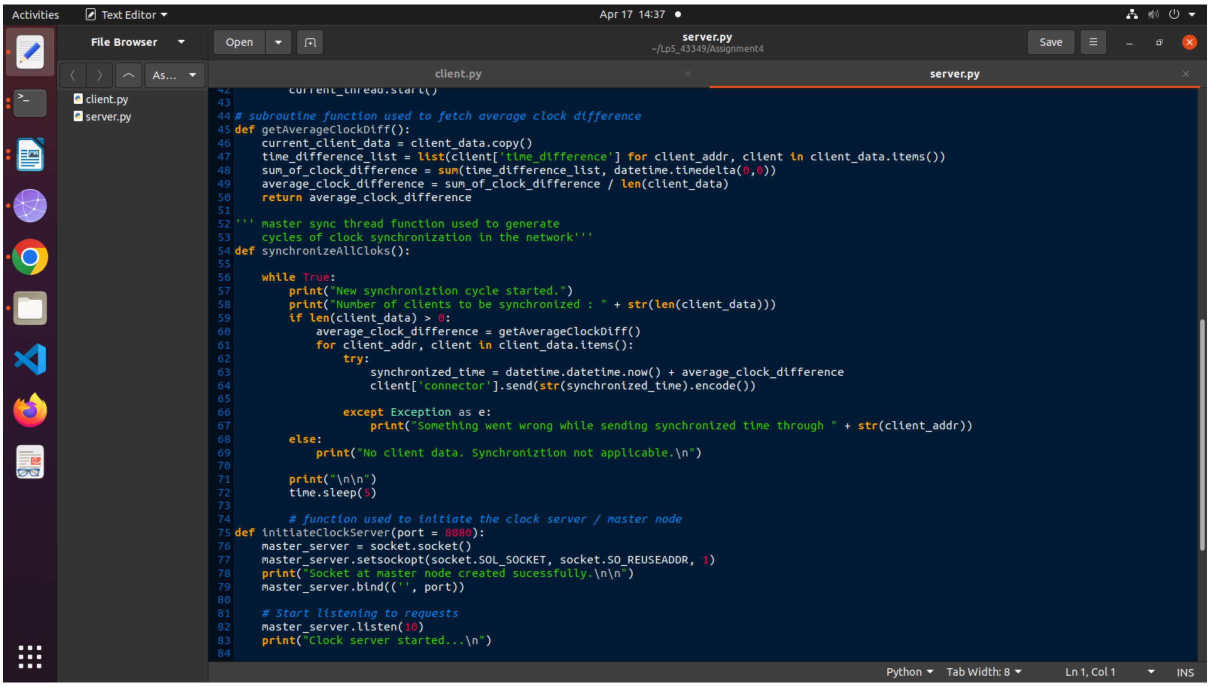
Assignment 4

Code-

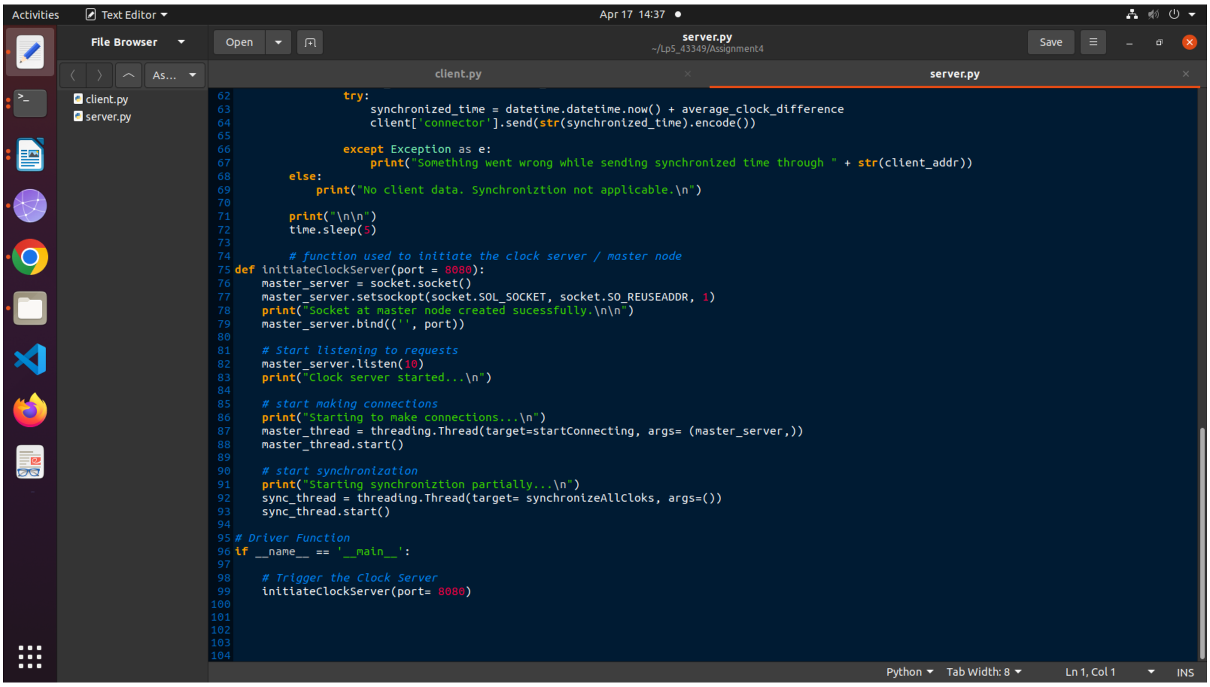
Server.py



```python
# Python3 program imitating a clock server

from functools import reduce
from dateutil import parser
import threading
import datetime
import socket
import time

# datastructure used to store client address and clock data
client_data = {}

''' nested thread function used to receive
    clock time from a connected client'''
def startReceivingClockTime(connector, address):

    while True:
        # receive clock time
        clock_time_string = connector.recv(1024).decode()
        clock_time = parser.parse(clock_time_string)
        clock_time_diff = datetime.datetime.now() - clock_time
        client_data[address] = {
            "clock_time" : clock_time,
            "time_difference" : clock_time_diff,
            "connector" : connector
        }
        print("Client Data updated with: "+ str(address), end="\n\n")
        time.sleep(5)


def startConnecting(master_server):

    # fetch clock time at slaves / clients
    while True:

        # accepting a client / slave clock client
        master_slave_connector, addr = master_server.accept()
        slave_address = str(addr[0]) + " : " + str(addr[1])
        print(slave_address + " got connected successfully.\n")

        current_thread = threading.Thread(target= startReceivingClockTime, args= (master_slave_connector, slave_address, ))
        current_thread.start()

```

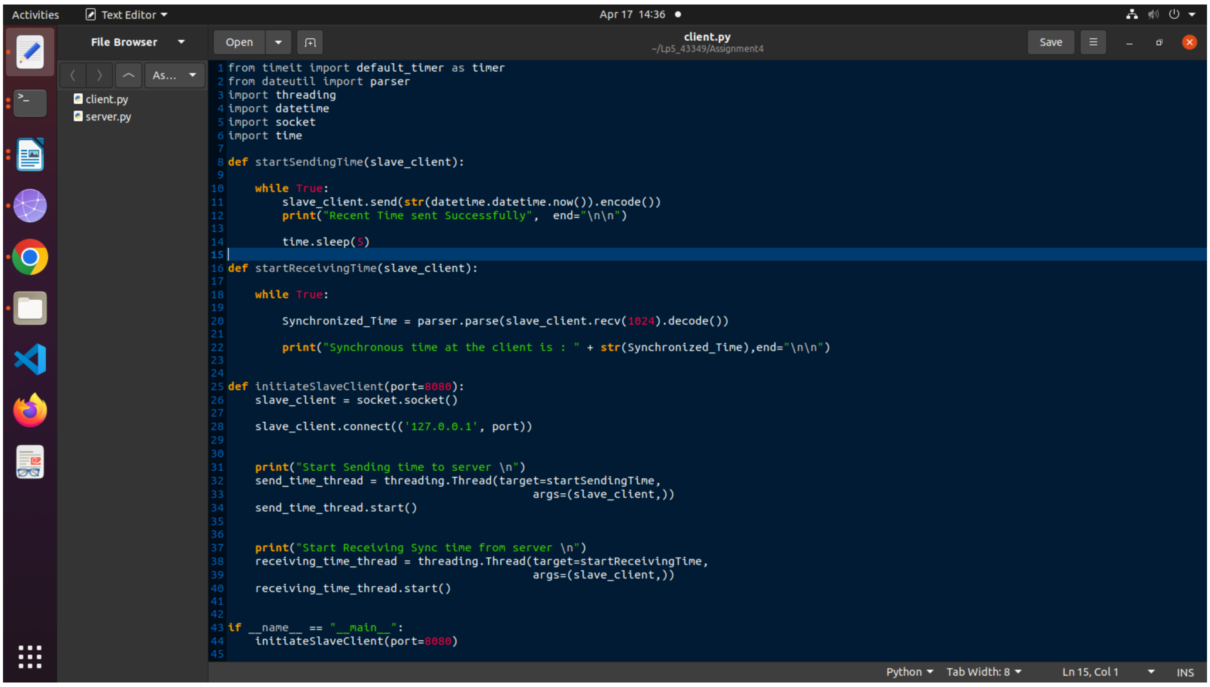


```python
        current_thread.start()

# subroutine function used to fetch average clock difference
def getAverageClockDiff():
    current_client_data = client_data.copy()
    time_difference_list = list(client['time_difference'] for client_addr, client in client_data.items())
    sum_of_clock_difference = sum(time_difference_list, datetime.timedelta(0,0))
    average_clock_difference = sum_of_clock_difference / len(client_data)
    return average_clock_difference

''' master sync thread function used to generate
    cycles of clock synchronization in the network'''
def synchronizeAllCloks():

    while True:
        print("New synchroniztion cycle started.")
        print("Number of clients to be synchronized : " + str(len(client_data)))
        if len(client_data) > 0:
            average_clock_difference = getAverageClockDiff()
            for client_addr, client in client_data.items():
                try:
                    synchronized_time = datetime.datetime.now() + average_clock_difference
                    client['connector'].send(str(synchronized_time).encode())

                except Exception as e:
                    print("Something went wrong while sending synchronized time through " + str(client_addr))
        else:
            print("No client data. Synchroniztion not applicable.\n")

        print("\n\n")
        time.sleep(5)

        # function used to initiate the clock server / master node
def initiateClockServer(port = 8080):
    master_server = socket.socket()
    master_server.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
    print("Socket at master node created sucessfully.\n\n")
    master_server.bind(('', port))

    # Start listening to requests
    master_server.listen(10)
    print("Clock server started...\n")

```



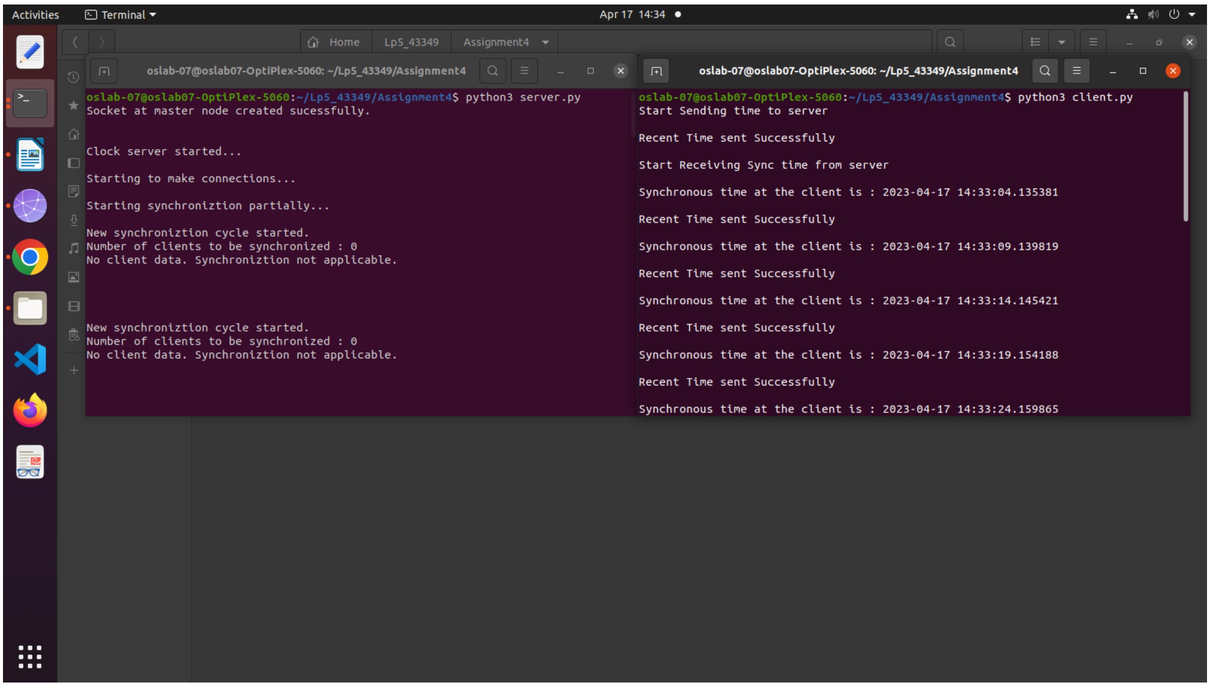
```
                try:
                    synchronized_time = datetime.datetime.now() + average_clock_difference
                    client['connector'].send(str(synchronized_time).encode())

                except Exception as e:
                    print("Something went wrong while sending synchronized time through " + str(client_addr))
        else:
            print("No client data. Synchroniztion not applicable.\n")

        print("\n\n")
        time.sleep(5)

        # function used to initiate the clock server / master node
def initiateClockServer(port = 8080):
    master_server = socket.socket()
    master_server.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
    print("Socket at master node created sucessfully.\n\n")
    master_server.bind(('', port))

    # Start listening to requests
    master_server.listen(10)
    print("Clock server started...\n")

    # start making connections
    print("Starting to make connections...\n")
    master_thread = threading.Thread(target=startConnecting, args= (master_server,))
    master_thread.start()

    # start synchronization
    print("Starting synchroniztion partially...\n")
    sync_thread = threading.Thread(target= synchronizeAllCloks, args=())
    sync_thread.start()

# Driver Function
if __name__ == '__main__':

    # Trigger the Clock Server
    initiateClockServer(port= 8080)
```

Client.py



```
from timeit import default_timer as timer
from dateutil import parser
import threading
import datetime
import socket
import time

def startSendingTime(slave_client):

    while True:
        slave_client.send(str(datetime.datetime.now()).encode())
        print("Recent Time sent Successfully",  end="\n\n")

        time.sleep(5)

def startReceivingTime(slave_client):

    while True:

        Synchronized_Time = parser.parse(slave_client.recv(1024).decode())

        print("Synchronous time at the client is : " + str(Synchronized_Time),end="\n\n")


def initiateSlaveClient(port=8080):
    slave_client = socket.socket()

    slave_client.connect(('127.0.0.1', port))


    print("Start Sending time to server \n")
    send_time_thread = threading.Thread(target=startSendingTime,
                                         args=(slave_client,))
    send_time_thread.start()


    print("Start Receiving Sync time from server \n")
    receiving_time_thread = threading.Thread(target=startReceivingTime,
                                         args=(slave_client,))
    receiving_time_thread.start()


if __name__ == "__main__":
    initiateSlaveClient(port=8080)
```

Output-


Activities
Terminal
Apr 17 14:34
Home
Lp5_43349
Assignment4
oslab-07@oslab07-OptiPlex-5060: ~/Lp5_43349/Assignment4
oslab-07@oslab07-OptiPlex-5060: ~/Lp5_43349/Assignment4

```
oslab-07@oslab07-OptiPlex-5060:~/Lp5_43349/Assignment4$ python3 server.py
Socket at master node created sucessfully.


Clock server started...

Starting to make connections...

Starting synchroniztion partially...

New synchroniztion cycle started.
Number of clients to be synchronized : 0
No client data. Synchroniztion not applicable.




New synchroniztion cycle started.
Number of clients to be synchronized : 0
No client data. Synchroniztion not applicable.
```

```
oslab-07@oslab07-OptiPlex-5060:~/Lp5_43349/Assignment4$ python3 client.py
Start Sending time to server

Recent Time sent Successfully

Start Receiving Sync time from server

Synchronous time at the client is : 2023-04-17 14:33:04.135381

Recent Time sent Successfully

Synchronous time at the client is : 2023-04-17 14:33:09.139819

Recent Time sent Successfully

Synchronous time at the client is : 2023-04-17 14:33:14.145421

Recent Time sent Successfully

Synchronous time at the client is : 2023-04-17 14:33:19.154188

Recent Time sent Successfully

Synchronous time at the client is : 2023-04-17 14:33:24.159865
```