Name: Mrunal Sanjay Chaudhari
Roll No: 47006 Class: BE-IT-B
Subject: Distributed Systems

Assignment No. 4

Problem Statement: Implement Berkeley algorithm for clock synchronization.

Code:

Client.py:

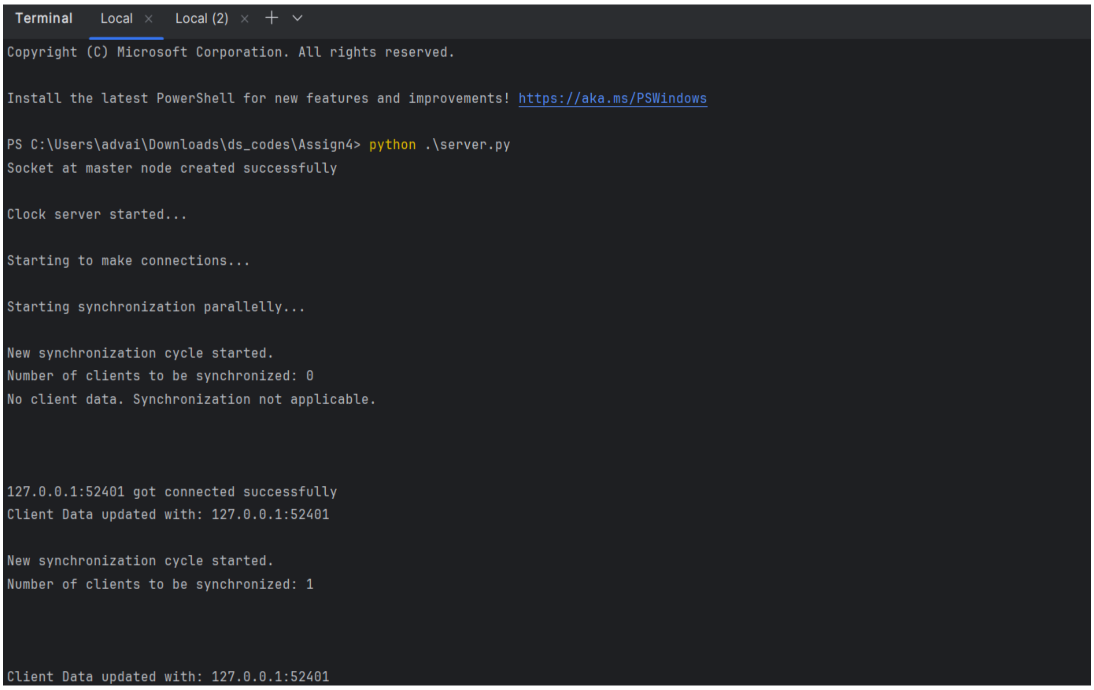
```
from timeit import default_timer as timer
from dateutil import parser
import threading
import datetime
import socket
import time
def startSendingTime(slave_client):
    while True:
        slave_client.send(str(
                    datetime.datetime.now()).encode())
        print("Recent time sent successfully",end = "\n\n")
        time.sleep(5)
def startReceivingTime(slave_client):
    while True:
        Synchronized_time = parser.parse(
                        slave_client.recv(1024).decode())
        print("Synchronized 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("Starting to receive time from server\n")
    send_time_thread = threading.Thread(target = startSendingTime,args = (slave_client, ))
    send_time_thread.start()
    print("Starting to receiving " + "synchronized time from server\n")
    receive_time_thread = threading.Thread(target = startReceivingTime,args = (slave_client, ))
    receive_time_thread.start()
if __name__ == '__main__':
    initiateSlaveClient(port = 8080)
```

Server.py:

```
from dateutil import parser
import threading
import datetime
import socket
import time
client_data = {}
def startReceivingClockTime(connector, address):
    while True:
        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):
    while True:
        master_slave_connector, addr = master_server.accept()
        slave_address = str(addr[0]) + ":" + str(addr[1])
        print(slave_address + " got connected successfully")
        current_thread = threading.Thread(target = startReceivingClockTime,args =
(master_slave_connector,slave_address, ))
        current_thread.start()
def getAverageClockDiff():
    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
def synchronizeAllClocks():
    while True:
        print("New synchronization 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." +" Synchronization not applicable.")
        print("\n\n")
        time.sleep(5)
def initiateClockServer(port = 8080):
    master_server = socket.socket()
    master_server.setsockopt(socket.SOL_SOCKET,socket.SO_REUSEADDR, 1)
    print("Socket at master node created successfully\n")
    master_server.bind(('', port))
    master_server.listen(10)
    print("Clock server started...\n")
    print("Starting to make connections...\n")
    master_thread = threading.Thread(target = startConnecting,args = (master_server, ))
    master_thread.start()
    print("Starting synchronization parallelly...\n")
    sync_thread = threading.Thread(target = synchronizeAllClocks,args = ())
    sync_thread.start()
if __name__ == '__main__':
    initiateClockServer(port = 8080)
```

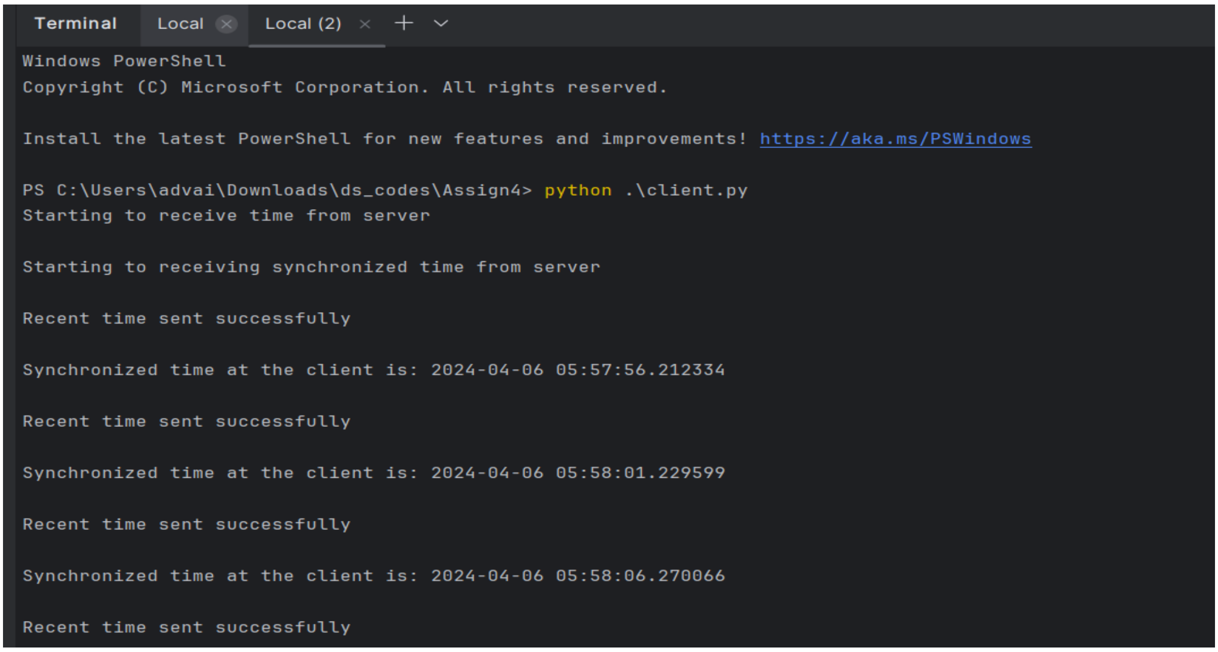
Output:

Server



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

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\advai\Downloads\ds_codes\Assign4> python .\server.py
Socket at master node created successfully

Clock server started...

Starting to make connections...

Starting synchronization parallelly...

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



127.0.0.1:52401 got connected successfully
Client Data updated with: 127.0.0.1:52401

New synchronization cycle started.
Number of clients to be synchronized: 1



Client Data updated with: 127.0.0.1:52401
```

Client



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

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\advai\Downloads\ds_codes\Assign4> python .\client.py
Starting to receive time from server

Starting to receiving synchronized time from server

Recent time sent successfully

Synchronized time at the client is: 2024-04-06 05:57:56.212334

Recent time sent successfully

Synchronized time at the client is: 2024-04-06 05:58:01.229599

Recent time sent successfully

Synchronized time at the client is: 2024-04-06 05:58:06.270066

Recent time sent successfully
```