

Assignment No. 4

Name: Viraj Sarjerao Kawade

Roll No: 043 BE-(IT)

Subject: LP-V (Distributed System)

Code:

```
1. Server

1. import threading
2. import datetime
3. import socket
4. import time
5. from dateutil import parser
6.
7. # Client data to be stored in variable client_data
8. client_data = {}
9.
10. # Nested thread function used to receive clock time from a connected client
11. def startReceivingClockTime(connector, address):
12.     while True:
13.         # receive clock time
14.         clock_time_string = connector.recv(1023).decode()
15.         clock_time = parser.parse(clock_time_string)
16.         clock_time_diff = datetime.datetime.now() - clock_time
17.         client_data[address] = {
18.             "clock_time": clock_time,
19.             "time_difference": clock_time_diff,
20.             "connector": connector
21.         }
22.         print("Client Date updated with: " + str(address), end="\n\n")
23.         time.sleep(5)
24.
25. # This opens up the master server to accept clients over given port
26. def startConnecting(master_server):
27.
28.     #fetching clock time at slaves / clients
29.     while True:
30.         master_slave_connector, addr = master_server.accept()
31.         slave_address = str(addr[0]) + ":" + str(addr[1])
32.         print(slave_address + " got connected successfully")
33.
34.         current_thread = threading.Thread(
35.             target = startReceivingClockTime,
36.             args = (master_slave_connector, slave_address,)
37.         )
38.         current_thread.start()
39.
40.
41. # Used to fetch average clock difference
42. def getAverageClockDiff():
43.     current_client_data = client_data.copy()
44.     time_difference_list = list(client["time_difference"] for client in
client_data.items())
45.     sum_of_clock_difference = sum(time_difference_list, datetime.timedelta(0,0))
46.     average_clock_difference = sum_of_clock_difference / len(client_data)
47.     return average_clock_difference
48.
49. # Master sync thread function used to generate cycles of clock synchronization in the
network
50. def synchronizeAllClocks():
51.     while True:
52.         print("New synchronization cycle started.")
53.         print("Number of clients to be synchronized: " + str(len(client_data)))
54.
55.         if len(client_data) > 0:
56.             average_clock_difference = getAverageClockDiff()
57.
58.             for client_addr, client in client_data.items():
59.                 try:
60.                     synchronized_time = datetime.datetime.now() + average_clock_difference
61.                     client["connector"].send(str(synchronized_time).encode())
62.
63.                 except Exception as e:
```

```

64.             print("Something went wring while sending synchronzied time
throughout"+str(client_addr))
65.         else:
66.             print("No client data. Synchronization not applicable.")
67.67.

68.         print("\n\n")
69.
70.         time.sleep(5)
71.
72. def initiateClockServer(port = 8080):
73.     master_server = socket.socket()
74.     master_server.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
75.
76.     print("Socket at master node created successfully")
77.
78.     HOST= "127.0.0.1"
79.     master_server.bind((HOST, port))
80.
81.     # Starts listening to requests
82.     master_server.listen(10)
83.     print("Clock server started...")
84.
85.     # start making connections
86.     print("Starting to make connections...")
87.     master_thread = threading.Thread(
88.         target = startConnecting,
89.         args = (master_server,)
90.     )
91.     master_thread.start()
92.
93.     # start synchronization
94.     print("Starting synchronization parallely...")
95.     sync_thread = threading.Thread(
96.         target = synchronizeAllClocks,
97.         args = ()
98.     )
99.     sync_thread.start()
100.
101.
102. # Driver function
103. if __name__ == '__main__':
104.     # Trigger the Clock Server
105.     initiateClockServer(port=8080)
106.

```

2. Client

```

1. from timeit import default_timer as timer
2. from dateutil import parser
3. import threading
4. import datetime
5. import socket
6. import time
7.
8.
9. # client thread function used to send time at client side
10. def startSendingTime(slave_client):
11.11.
12.     while True:
13.         # provide server with clock time at the client
14.         slave_client.send(str(datetime.datetime.now()).encode())
15.
16.         print("Recent time sent successfully", end = "\n\n")
17.         time.sleep(5)
18.
19.

```

```
20. # client thread function used to receive synchronized time
21. def startReceivingTime(slave_client):
22.22.
23.     while True:
24.         # receive data from the server
25.         Synchronized_time = parser.parse(slave_client.recv(1024).decode())
26.
27.         print("Synchronized time at the client is: " + str(Synchronized_time), end = "\n\n")
28.
29.
30. # function used to Synchronize client process time
31. def initiateSlaveClient(port = 8080):
32.32.
33.     slave_client = socket.socket()
34.
35.     # connect to the clock server on local computer
36.     slave_client.connect(('127.0.0.1', port))
37.
38.     # start sending time to server
39.     print("Starting to receive time from server\n")
40.     send_time_thread = threading.Thread(
41.         target = startSendingTime,
42.         args = (slave_client, ))
43.     send_time_thread.start()
44.
45.
46.     # start receiving synchronized from server
47.     print("Starting to receiving " + "synchronized time from server\n")
48.     receive_time_thread = threading.Thread(
49.         target = startReceivingTime,
50.         args = (slave_client, ))
51.     receive_time_thread.start()
52.
53.
54. # Driver function
55. if __name__ == ' main ':
56.
57.     # initialize the Slave / Client
58.     initiateSlaveClient(port = 8080)
59.
```

Output:

1. Start the Server:

```
Command Prompt - python server.py
Microsoft Windows [Version 10.0.19045.2965]
(c) Microsoft Corporation. All rights reserved.

C:\Users\hp>cd C:\Users\hp\Desktop\ds lab\Assignment No.4

C:\Users\hp\Desktop\ds lab\Assignment No.4>python server.py
Socket at master node created successfully
Clock server started...
Starting to make connections...
Starting synchronization in parallel...
New synchronization cycle started.
Number of clients to be synchronized: 0
No client data. Synchronization not applicable.

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

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

2.Start client:

```
Command Prompt - python client.py
Microsoft Windows [Version 10.0.19045.2965]
(c) Microsoft Corporation. All rights reserved.

C:\Users\hp>cd C:\Users\hp\Desktop\ds lab\Assignment No.4

C:\Users\hp\Desktop\ds lab\Assignment No.4>python client.py
Starting to receive time from the server

Starting to receive synchronized time from the server

Recent time sent successfully

Synchronized time at the client is: 2023-05-24 18:39:11.877943
```

3.Server After Synchronization:

```
Command Prompt - python server.py

127.0.0.1:58394 got connected successfully
Client data updated with: 127.0.0.1:58394

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

Client data updated with: 127.0.0.1:58394

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

Client data updated with: 127.0.0.1:58394

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

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