GOA COLLEGE OF ENGINEERING

“Bhausaheb Bandodkar Technical Education Complex”

Experiment No: 4

Clock Synchronization

Aim: To implement Clock synchronization in Distributed Operating systems.

Theory:

Each node in distributed systems can share their resources with other nodes. So, there is need of proper allocation of resources to preserve the state of resources and help coordinate between the several processes. To resolve such conflicts, synchronization is used. Synchronization in distributed systems is achieved via clocks.

The physical clocks are used to adjust the time of nodes. Each node in the system can share its local time with other nodes in the system. The time is set based on UTC (Universal Time Coordination). UTC is used as a reference time clock for the nodes in the system.

The clock synchronization can be achieved by 2 ways: External and Internal Clock Synchronization.

1. **External clock synchronization** is the one in which an external reference clock is present. It is used as a reference and the nodes in the system can set and adjust their time accordingly.

2. **Internal clock synchronization** is the one in which each node shares its time with other nodes and all the nodes set and adjust their times accordingly.

There are 2 types of clock synchronization algorithms: Centralized and Distributed.

1. **Centralized** is the one in which a time server is used as a reference. The single time server propagates it’s time to the nodes and all the nodes adjust the time accordingly. It is dependent on single time server so if that node fails, the whole system will lose synchronization. Examples of centralized are- Berkeley Algorithm, Passive Time Server, Active Time Server etc.

2. **Distributed** is the one in which there is no centralized time server present. Instead, the nodes adjust their time by using their local time and then, taking the average of the differences of time with other nodes. Distributed algorithms overcome the issue of centralized algorithms like the scalability and single point failure. Examples of Distributed algorithms are – Global Averaging Algorithm, Localized Averaging Algorithm, NTP (Network time protocol) etc.

Program:
//server code

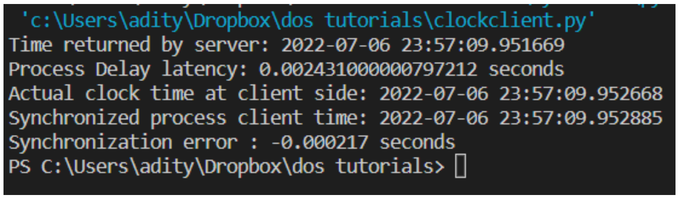
```
# Python3 program imitating a clock server
import socket
import datetime
# function used to initiate the clock Server
def initiateclockserver():
    s = socket.socket()
    print("Socket successfully created")

    # Server port
    port = 8000
    s.bind(('', port))


    # Start listening to requests
    s.listen (5)
    print("Socket is listening...")

    # clock Server Running forever
    while True:
    # Establish connection with client
        connection, address = s.accept()
        print('server connected to', address)
        # Respond the client with server clock time
        connection.send(str(datetime.datetime.now()).encode())
        # close the connection with the client process
        connection.close()
# Driver function
if __name__ == '__main__':
    # Trigger the clock Server
    initiateclockserver()
```

//client code

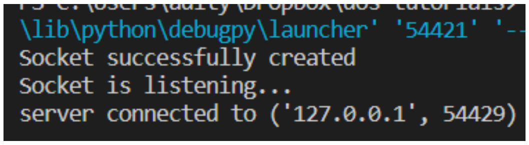
```
import socket
import datetime
```

```
from dateutil import parser
from timeit import default_timer as timer
# function used to Synchronize client process time
def synchronizeTime():
    s = socket.socket()
    # Server port
    port = 8000
    # connect to the clock server on local computer
    s.connect(('127.0.0.1', port))
    request_time = timer()
    # receive data from the server
    server_time = parser.parse(s.recv(1024).decode())
    response_time = timer()
    actual_time = datetime.datetime.now()
    print("Time returned by server: " + str(server_time))
    process_delay_latency = response_time - request_time
    print("Process Delay latency: "\
            + str(process_delay_latency) \
            + " seconds")
    print("Actual clock time at client side: "\
            + str(actual_time))
    # synchronize process client clock time
    client_time = server_time + datetime.timedelta(seconds = (process_delay_latency) / 2)
    print("Synchronized process client time: " + str(client_time))
    # calculate synchronization error
    error = actual_time - client_time
    print("Synchronization error : " + str(error.total_seconds()) + " seconds")
    s.close()
# Driver function
if __name__ == '__main__':
    # Trigger the clock Server
    synchronizeTime()
```

Output:



```
 'c:\Users\adity\Dropbox\dos tutorials\clockclient.py'
Time returned by server: 2022-07-06 23:57:09.951669
Process Delay latency: 0.002431000000797212 seconds
Actual clock time at client side: 2022-07-06 23:57:09.952668
Synchronized process client time: 2022-07-06 23:57:09.952885
Synchronization error : -0.000217 seconds
PS C:\Users\adity\Dropbox\dos tutorials> 
```



```
PS C:\Users\adity\Dropbox\dos tutorials>
 \lib\python\debugpy\launcher' '54421' '--
 Socket successfully created
 Socket is listening...
 server connected to ('127.0.0.1', 54429)
```

Conclusion: Clock Synchronization experiment was implemented and executed successfully.