

Introduction:

This project assignment two mainly focused on clock synchronization where the clocks on the clients and the server are synchronized. I have used socket programming for the synchronization of clocks between the server and the clients. Clock synchronization is performed using the famous Berkeley algorithm.

Client side:

The client side basically receives the request sent by the server . The request is to send the local time on the client machine. The logical clock time on the client's local machine has been randomly initialized to some value by using the function `rand()`. This function randomly initializes the logical clock of multiple clients during the execution. This clock value from all the clients are sent to the server when it requests the values from the clients.

Server Side:

The Server/Master is the one that starts first, it sends the initial request to the client asking for its clock values. The connection that I have used for communication is TCP. Multiple sockets have been created by having the `accept` in the server side to be iterative. Thus according to the number of clients , the sockets `accept()` is generated and used. The Server after receiving the clock time from the client sends ,it takes the average of all the slave/clients along with the master. The average value is sent as the result to all the clients simultaneously .

Implementation:

The aim of this project is to synchronize clocks among the server and the clients. The clock value of the client is initialized randomly. The same way the server/master's clock values has been initialized randomly. The Server starts first and initiates the clients by sending the message to "send the clock time ".The clients after receiving the message sent by the server would send the clock value which is randomly initialized.

Based on the number of clients , the server creates new socket for the clients data to be received.

Eg:

```
New_socket[i]=accept(server_fd,(struct sockaddr*)&address,(socklen_t*)&addrlen);
```

By the above command ,the connection made for multiple clients. In the same way, the data is received for the multiple clients and the Berkeley algorithm is applied by taking average of the clock values of the client and the server. The resulting value is sent to the clients.

The clients are executed simultaneously using a python code where the number of client process is received as an argument and executes the client process through a module called `subprocess` and `multiprocess`.

Execution:

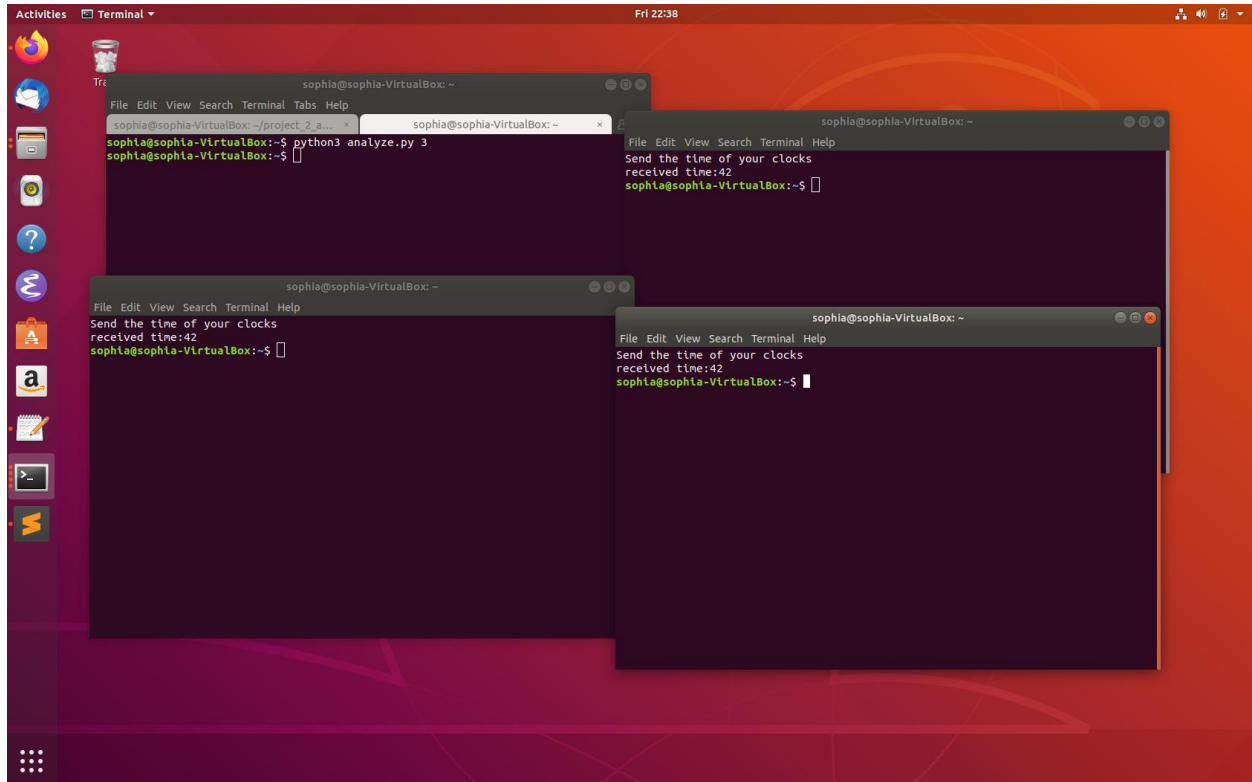
`make compile`

`./master 3`

`python3 analyze.py 3`

Output:

I have displayed the outputs of each and every client in the separate command window.

**Result:**

Thus, I have implemented the clock synchronization among the server and multiple clients using Berkeley algorithm.