CMSC 621
Advanced Operating Systems
Project-2
JA52979
SAI TEJA CHALLA
REPORT

Clock, Multicast, and COMMIT

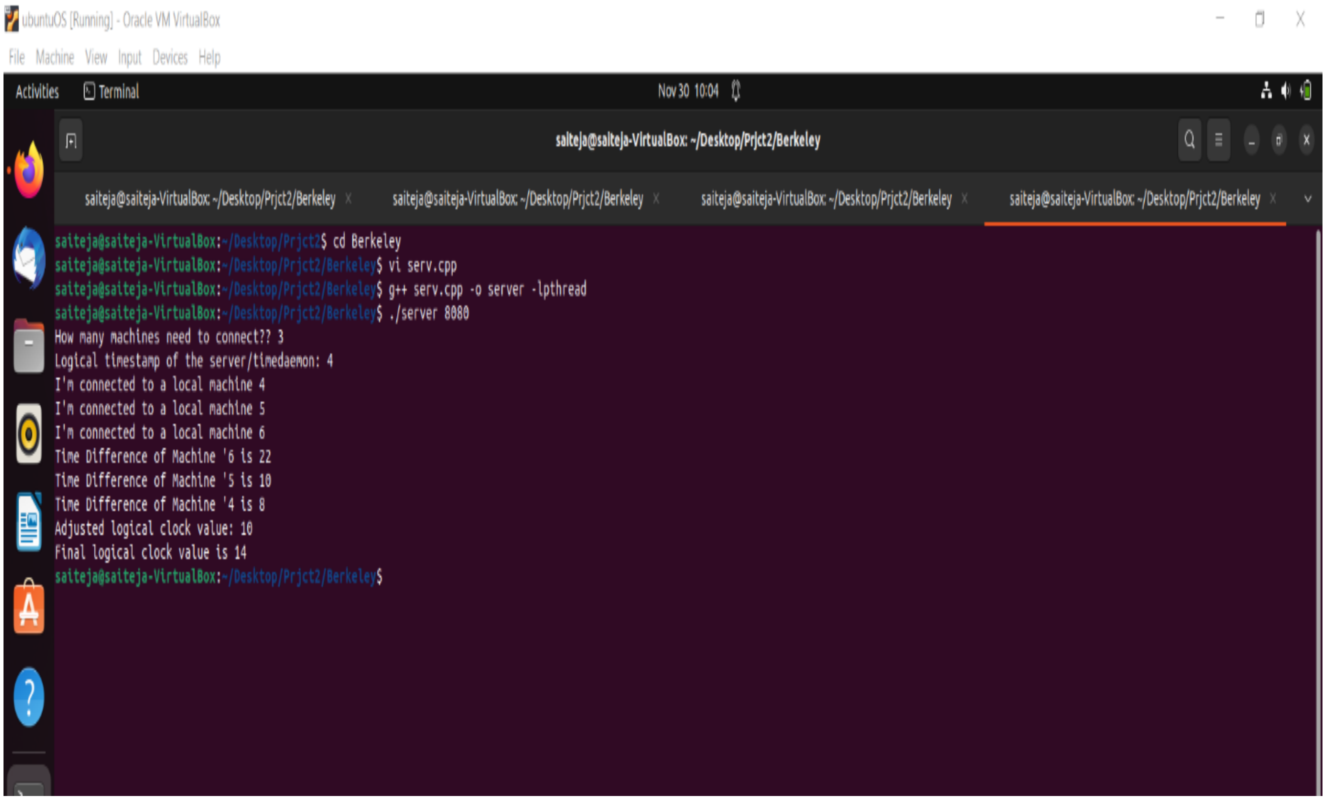
Assignment-1:

For this assignment, a client-server architecture is built to implement Berkeley's Algorithm.

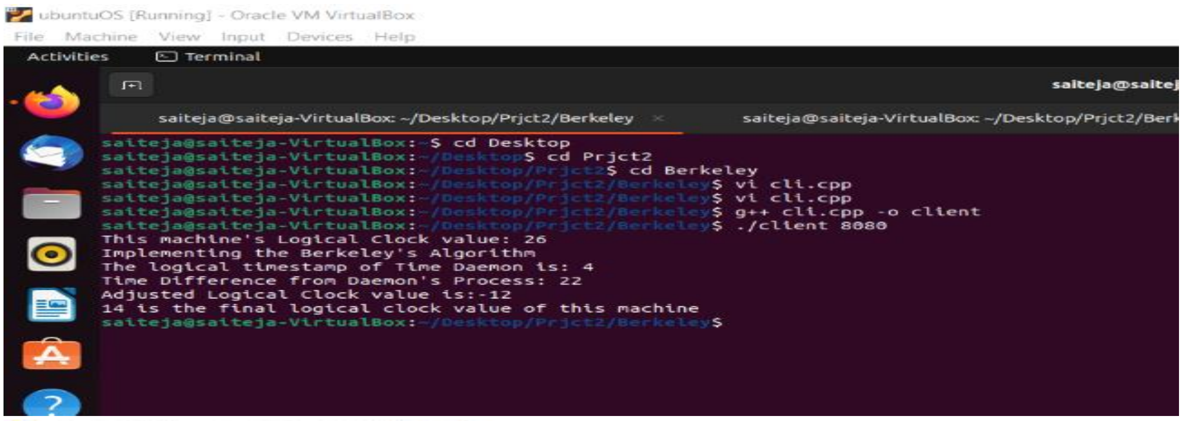
Here, the server (Time Daemon) keeps accepting all the connections from the desired machines. For each connection with a client, a new thread is created for communication. Using the "rand()" function, the server, and all the client machines initialize their logical clock with a random value in the range of 0 to 50 integer values.

The clients then compute the difference between their own logical clocks and the logical clock of the Time Daemon and communicate this time difference back to the server. The server calculates the average value by adding the time differences of all connected clients and its own time difference (which is always zero). The server determines the time adjustment for each client as follows: (average - time difference) and delivers this time adjustment back to the client. All of the clients receive their respective adjustment values and add them to their logical clocks. The Time Daemon adds the average value to its logical clock, and the algorithm ends.

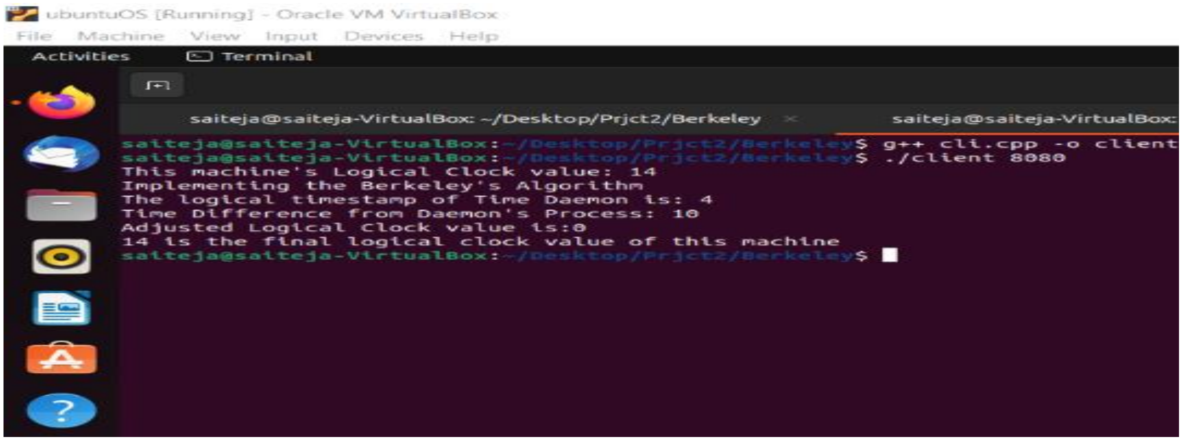
Output:


ubuntuOS [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
Terminal
Nov 30 10:04
saiteja@saiteja-VirtualBox: ~/Desktop/Prjct2/Berkeley

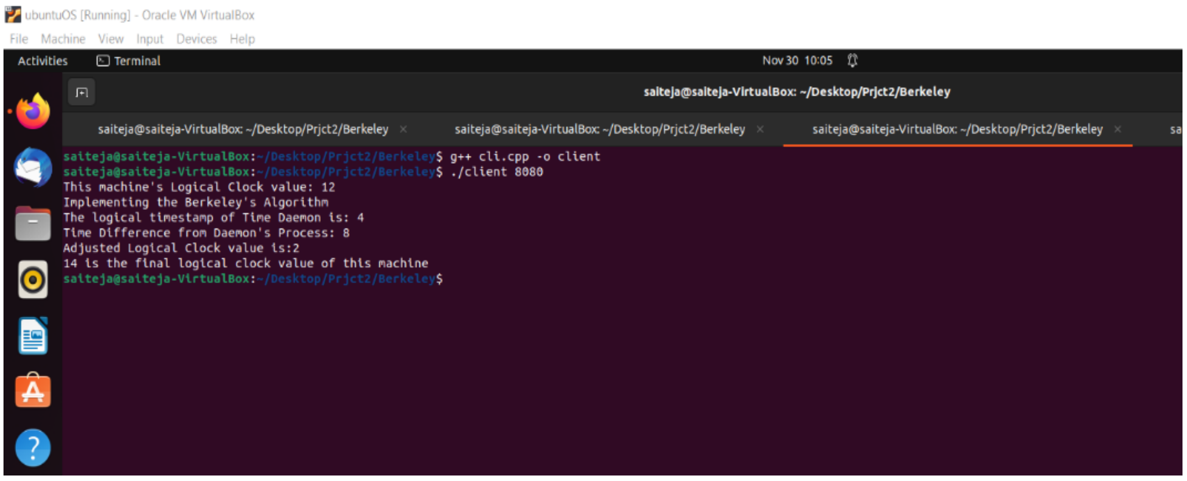
```
saiteja@saiteja-VirtualBox:~/Desktop/Prjct2$ cd Berkeley
saiteja@saiteja-VirtualBox:~/Desktop/Prjct2/Berkeley$ vi serv.cpp
saiteja@saiteja-VirtualBox:~/Desktop/Prjct2/Berkeley$ g++ serv.cpp -o server -lpthread
saiteja@saiteja-VirtualBox:~/Desktop/Prjct2/Berkeley$ ./server 8080
How many machines need to connect?? 3
Logical timestamp of the server/timedaemon: 4
I'm connected to a local machine 4
I'm connected to a local machine 5
I'm connected to a local machine 6
Time Difference of Machine '6 is 22
Time Difference of Machine '5 is 10
Time Difference of Machine '4 is 8
Adjusted logical clock value: 10
Final logical clock value is 14
saiteja@saiteja-VirtualBox:~/Desktop/Prjct2/Berkeley$
```


ubuntuOS [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
Terminal
saiteja@saiteja-VirtualBox: ~/Desktop/Prjct2/Berkeley

```
saiteja@saiteja-VirtualBox:~$ cd Desktop
saiteja@saiteja-VirtualBox:~/Desktop$ cd Prjct2
saiteja@saiteja-VirtualBox:~/Desktop/Prjct2$ cd Berkeley
saiteja@saiteja-VirtualBox:~/Desktop/Prjct2/Berkeley$ vi cli.cpp
saiteja@saiteja-VirtualBox:~/Desktop/Prjct2/Berkeley$ vi cli.cpp
saiteja@saiteja-VirtualBox:~/Desktop/Prjct2/Berkeley$ g++ cli.cpp -o client
saiteja@saiteja-VirtualBox:~/Desktop/Prjct2/Berkeley$ ./client 8080
This machine's Logical Clock value: 26
Implementing the Berkeley's Algorithm
The logical timestamp of Time Daemon is: 4
Time Difference from Daemon's Process: 22
Adjusted Logical Clock value is:-12
14 is the final logical clock value of this machine
saiteja@saiteja-VirtualBox:~/Desktop/Prjct2/Berkeley$
```


ubuntuOS [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
Terminal
saiteja@saiteja-VirtualBox: ~/Desktop/Prjct2/Berkeley

```
saiteja@saiteja-VirtualBox:~/Desktop/Prjct2/Berkeley$ g++ cli.cpp -o client
saiteja@saiteja-VirtualBox:~/Desktop/Prjct2/Berkeley$ ./client 8080
This machine's Logical Clock value: 14
Implementing the Berkeley's Algorithm
The logical timestamp of Time Daemon is: 4
Time Difference from Daemon's Process: 10
Adjusted Logical Clock value is:0
14 is the final logical clock value of this machine
saiteja@saiteja-VirtualBox:~/Desktop/Prjct2/Berkeley$
```


ubuntuOS [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
Terminal
Nov 30 10:05
saiteja@saiteja-VirtualBox: ~/Desktop/Prjct2/Berkeley

```
saiteja@saiteja-VirtualBox:~/Desktop/Prjct2/Berkeley$ g++ cli.cpp -o client
saiteja@saiteja-VirtualBox:~/Desktop/Prjct2/Berkeley$ ./client 8080
This machine's Logical Clock value: 12
Implementing the Berkeley's Algorithm
The logical timestamp of Time Daemon is: 4
Time Difference from Daemon's Process: 8
Adjusted Logical Clock value is:2
14 is the final logical clock value of this machine
saiteja@saiteja-VirtualBox:~/Desktop/Prjct2/Berkeley$
```

By implementing this, one can get to know about how client-server architecture and how the Berkeley Algorithm works in a distributed network.

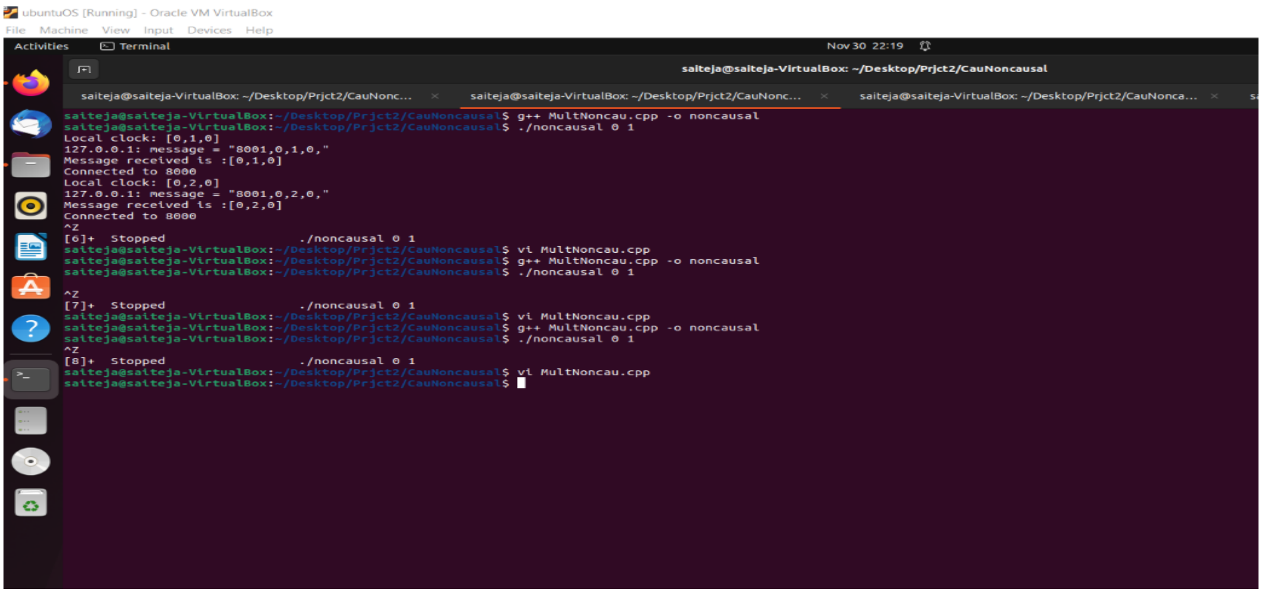
While implementing this, one can face issues in the usage of port numbers and IP addresses. To overcome that, we can make use of “setsockopt()” function, also to avoid confusion in calculating the time differences across all the clients, we can make sure that the server waits for all connections to complete before starting the algorithm and wait till all the clients have sent their time differences before calculating the average value.

Assignment-2:

Limited to three processes.

Non-Causal:

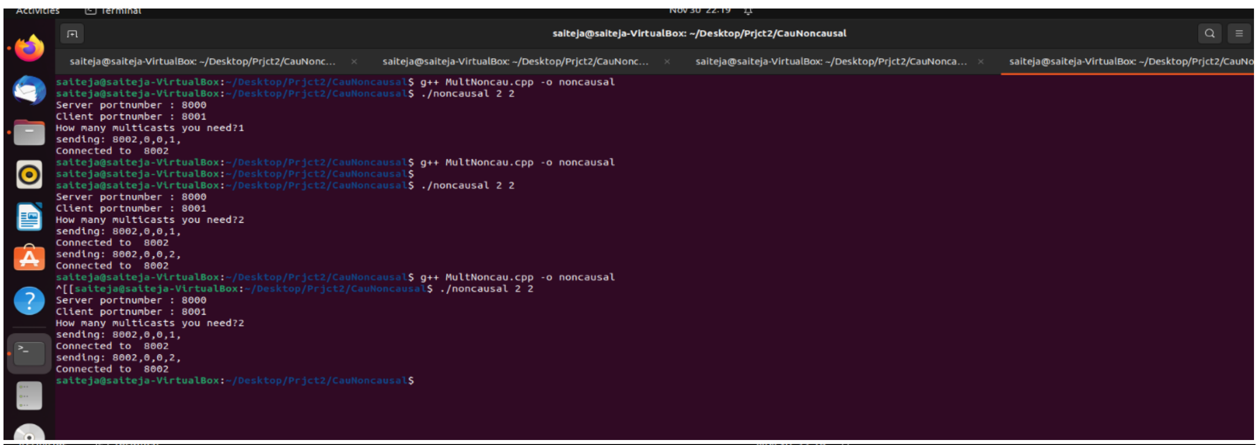
For this assignment, two functions have implemented a client and a server. Here, the processes are limited to 3. The local time is sent to the server, whatever the message received by the server, the vector clock is updated at the server.



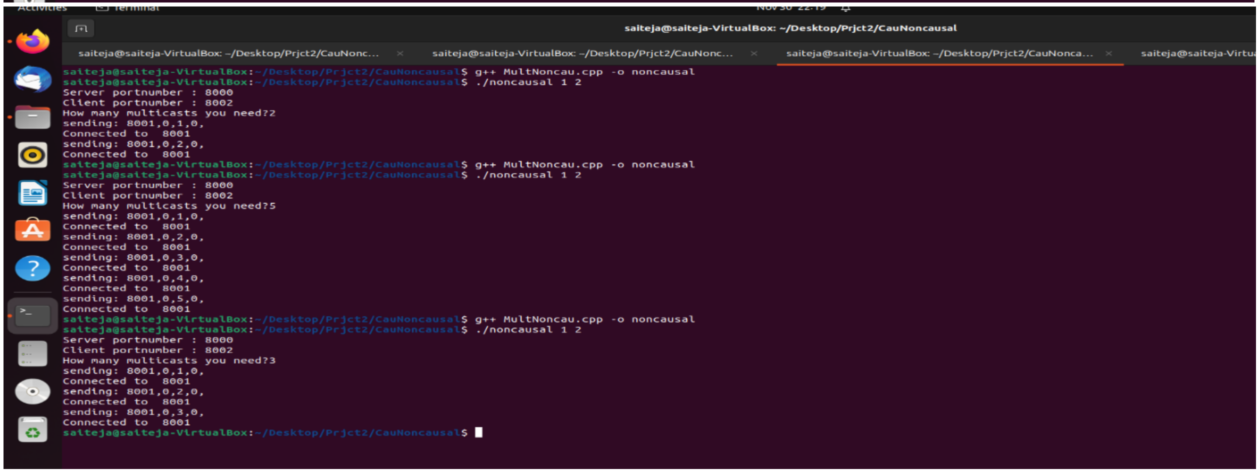
ubuntuOS [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
Terminal
Nov 30 22:19
saiteja@saiteja-VirtualBox: ~/Desktop/Prjct2/CauNoncausal

```
saiteja@saiteja-VirtualBox:~/Desktop/Prjct2/CauNoncausal$ g++ MultNoncau.cpp -o noncausal
saiteja@saiteja-VirtualBox:~/Desktop/Prjct2/CauNoncausal$ ./noncausal 0 1
Local clock: [0,1,0]
127.0.0.1: message = "8001,0,1,0,"
Message received is :[0,1,0]
Connected to 8000
Local clock: [0,2,0]
127.0.0.1: message = "8001,0,2,0,"
Message received is :[0,2,0]
Connected to 8000
^Z
[6]+  Stopped                 ./noncausal 0 1
saiteja@saiteja-VirtualBox:~/Desktop/Prjct2/CauNoncausal$ vi MultNoncau.cpp
saiteja@saiteja-VirtualBox:~/Desktop/Prjct2/CauNoncausal$ g++ MultNoncau.cpp -o noncausal
saiteja@saiteja-VirtualBox:~/Desktop/Prjct2/CauNoncausal$ ./noncausal 0 1

^Z
[7]+  Stopped                 ./noncausal 0 1
saiteja@saiteja-VirtualBox:~/Desktop/Prjct2/CauNoncausal$ vi MultNoncau.cpp
saiteja@saiteja-VirtualBox:~/Desktop/Prjct2/CauNoncausal$ g++ MultNoncau.cpp -o noncausal
saiteja@saiteja-VirtualBox:~/Desktop/Prjct2/CauNoncausal$ ./noncausal 0 1
^Z
[8]+  Stopped                 ./noncausal 0 1
saiteja@saiteja-VirtualBox:~/Desktop/Prjct2/CauNoncausal$ vi MultNoncau.cpp
saiteja@saiteja-VirtualBox:~/Desktop/Prjct2/CauNoncausal$ 
```

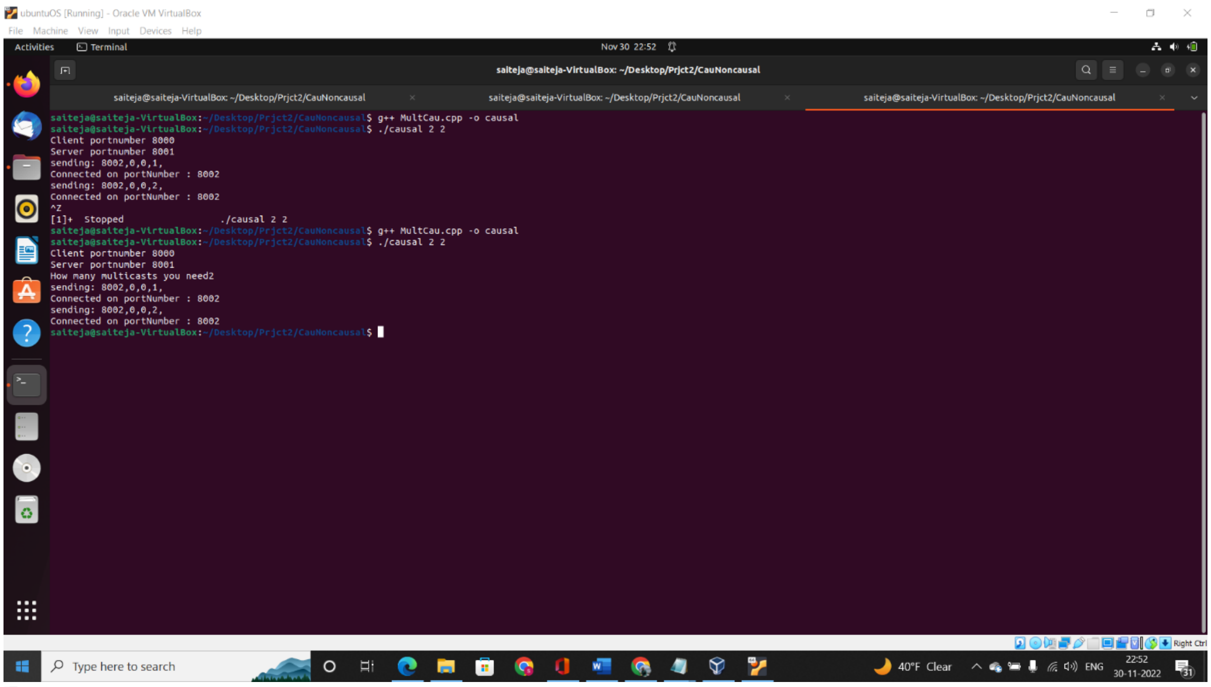


```
saiteja@saiteja-VirtualBox:~/Desktop/Prjct2/CauNoncausal$ g++ MultNoncau.cpp -o noncausal
saiteja@saiteja-VirtualBox:~/Desktop/Prjct2/CauNoncausal$ ./noncausal 2 2
Server portnumber : 8000
Client portnumber : 8001
How many multicasts you need?1
sending: 8002,0,0,1,
Connected to  8002
saiteja@saiteja-VirtualBox:~/Desktop/Prjct2/CauNoncausal$ g++ MultNoncau.cpp -o noncausal
saiteja@saiteja-VirtualBox:~/Desktop/Prjct2/CauNoncausal$
saiteja@saiteja-VirtualBox:~/Desktop/Prjct2/CauNoncausal$ ./noncausal 2 2
Server portnumber : 8000
Client portnumber : 8001
How many multicasts you need?2
sending: 8002,0,0,1,
Connected to  8002
sending: 8002,0,0,2,
Connected to  8002
saiteja@saiteja-VirtualBox:~/Desktop/Prjct2/CauNoncausal$ g++ MultNoncau.cpp -o noncausal
^[[saiteja@saiteja-VirtualBox:~/Desktop/Prjct2/CauNoncausal$ ./noncausal 2 2
Server portnumber : 8000
Client portnumber : 8001
How many multicasts you need?2
sending: 8002,0,0,1,
Connected to  8002
sending: 8002,0,0,2,
Connected to  8002
saiteja@saiteja-VirtualBox:~/Desktop/Prjct2/CauNoncausal$
```

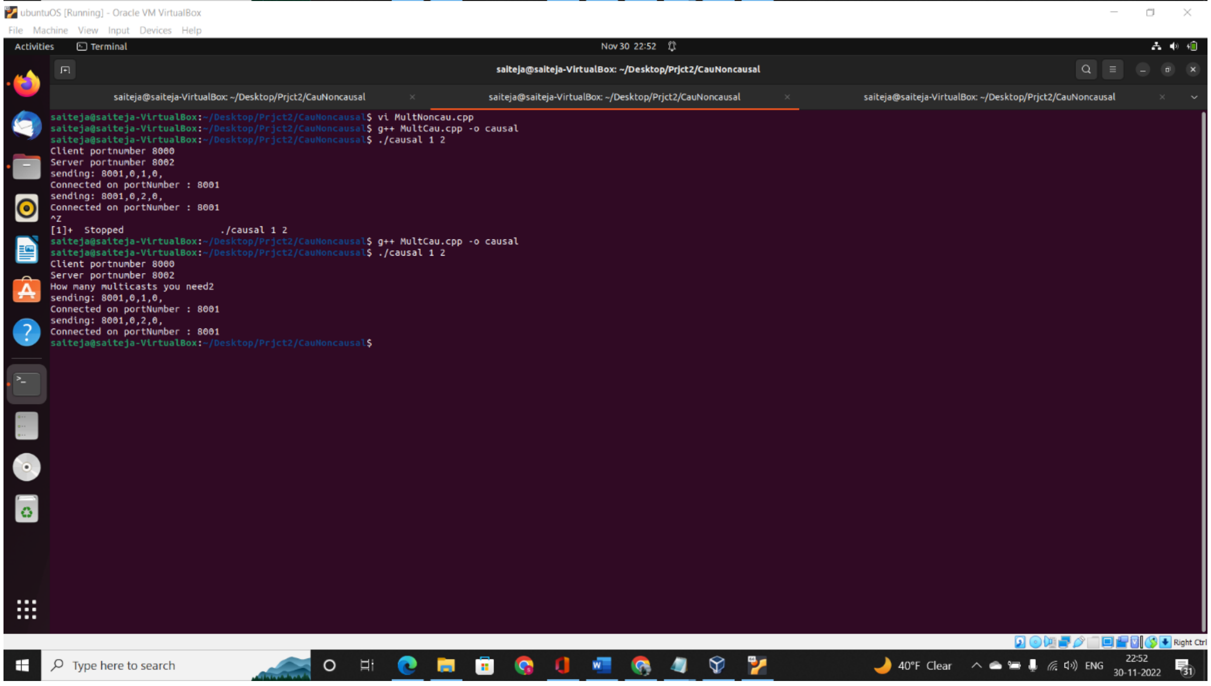


```
saiteja@saiteja-VirtualBox:~/Desktop/Prjct2/CauNoncausal$ g++ MultNoncau.cpp -o noncausal
saiteja@saiteja-VirtualBox:~/Desktop/Prjct2/CauNoncausal$ ./noncausal 1 2
Server portnumber : 8000
Client portnumber : 8002
How many multicasts you need?2
sending: 8001,0,1,0,
Connected to  8001
sending: 8001,0,2,0,
Connected to  8001
saiteja@saiteja-VirtualBox:~/Desktop/Prjct2/CauNoncausal$ g++ MultNoncau.cpp -o noncausal
saiteja@saiteja-VirtualBox:~/Desktop/Prjct2/CauNoncausal$ ./noncausal 1 2
Server portnumber : 8000
Client portnumber : 8002
How many multicasts you need?5
sending: 8001,0,1,0,
Connected to  8001
sending: 8001,0,2,0,
Connected to  8001
sending: 8001,0,3,0,
Connected to  8001
sending: 8001,0,4,0,
Connected to  8001
sending: 8001,0,5,0,
Connected to  8001
saiteja@saiteja-VirtualBox:~/Desktop/Prjct2/CauNoncausal$ g++ MultNoncau.cpp -o noncausal
saiteja@saiteja-VirtualBox:~/Desktop/Prjct2/CauNoncausal$ ./noncausal 1 2
Server portnumber : 8000
Client portnumber : 8002
How many multicasts you need?3
sending: 8001,0,1,0,
Connected to  8001
sending: 8001,0,2,0,
Connected to  8001
sending: 8001,0,3,0,
Connected to  8001
saiteja@saiteja-VirtualBox:~/Desktop/Prjct2/CauNoncausal$ 
```

Causal:



```
saiteja@saiteja-VirtualBox:~/Desktop/Prjct2/CauNoncausal$ g++ MultCau.cpp -o causal
saiteja@saiteja-VirtualBox:~/Desktop/Prjct2/CauNoncausal$ ./causal 2 2
Client portnumber 8000
Server portnumber 8001
sending: 8002,0,0,1,
Connected on portNumber : 8002
sending: 8002,0,0,2,
Connected on portNumber : 8002
^Z
[1]+  Stopped                 ./causal 2 2
saiteja@saiteja-VirtualBox:~/Desktop/Prjct2/CauNoncausal$ g++ MultCau.cpp -o causal
saiteja@saiteja-VirtualBox:~/Desktop/Prjct2/CauNoncausal$ ./causal 2 2
Client portnumber 8000
Server portnumber 8001
How many multicasts you need2
sending: 8002,0,0,1,
Connected on portNumber : 8002
sending: 8002,0,0,2,
Connected on portNumber : 8002
saiteja@saiteja-VirtualBox:~/Desktop/Prjct2/CauNoncausal$ 
```



```
saiteja@saiteja-VirtualBox:~/Desktop/Prjct2/CauNoncausal$ vi MultNoncau.cpp
saiteja@saiteja-VirtualBox:~/Desktop/Prjct2/CauNoncausal$ g++ MultCau.cpp -o causal
saiteja@saiteja-VirtualBox:~/Desktop/Prjct2/CauNoncausal$ ./causal 1 2
Client portnumber 8000
Server portnumber 8002
sending: 8001,0,1,0,
Connected on portNumber : 8001
sending: 8001,0,2,0,
Connected on portNumber : 8001
^Z
[1]+  Stopped                 ./causal 1 2
saiteja@saiteja-VirtualBox:~/Desktop/Prjct2/CauNoncausal$ g++ MultCau.cpp -o causal
saiteja@saiteja-VirtualBox:~/Desktop/Prjct2/CauNoncausal$ ./causal 1 2
Client portnumber 8000
Server portnumber 8002
How many multicasts you need2
sending: 8001,0,1,0,
Connected on portNumber : 8001
sending: 8001,0,2,0,
Connected on portNumber : 8001
saiteja@saiteja-VirtualBox:~/Desktop/Prjct2/CauNoncausal$
```

Here, the same is implemented , but the buffered messages are received and updated at the server side.

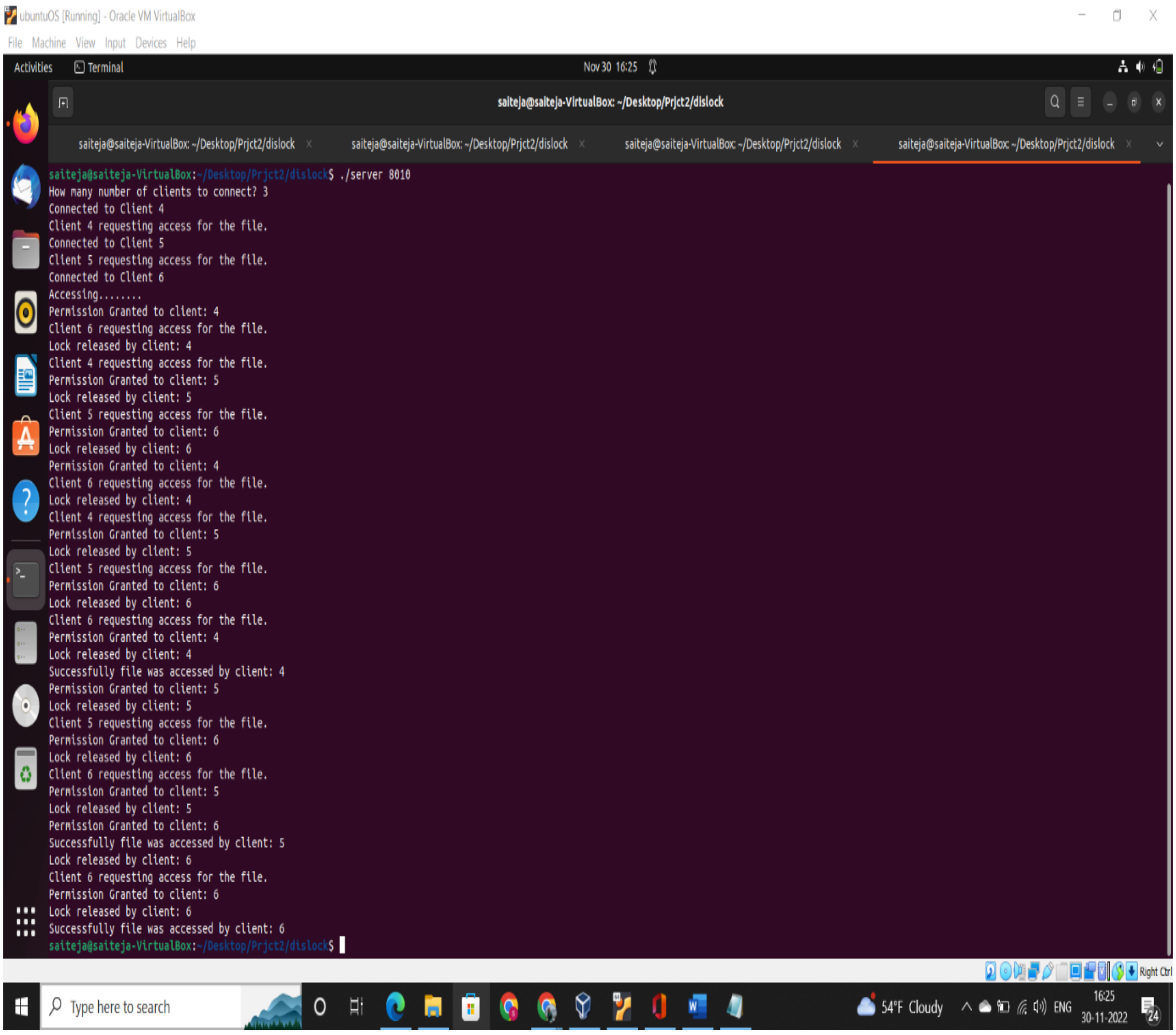
By implementing this one can able to build a code containing the server and client in the same code.

Bonus Assignment:

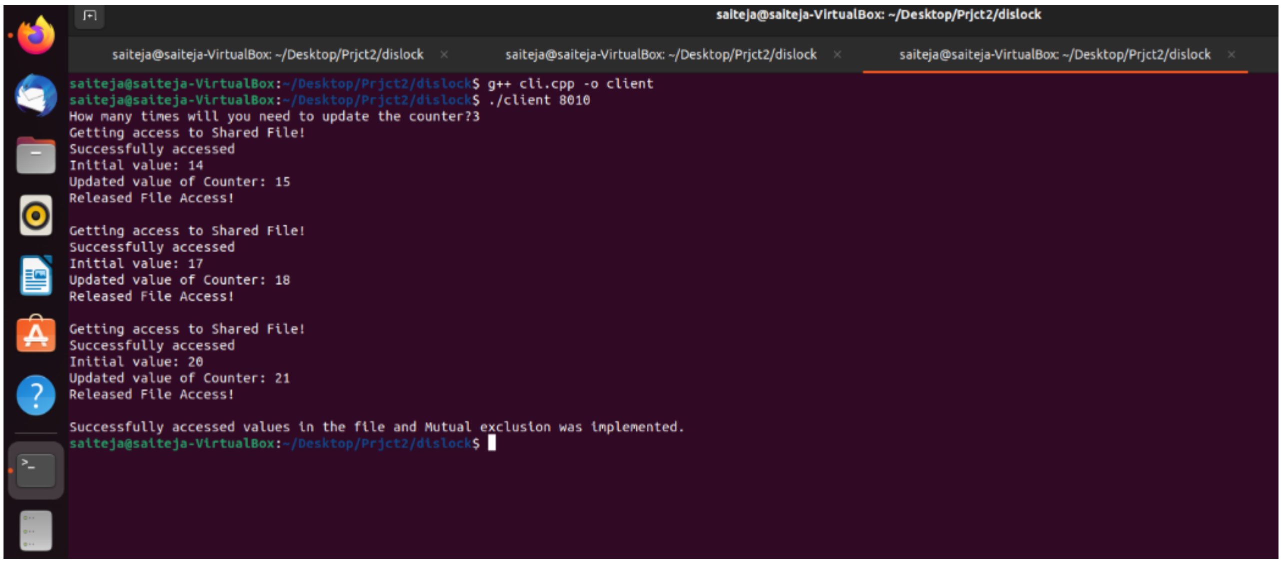
A client-server architecture is built to access a shared file using the locking mechanism. Here also the same is implemented for client-server communication. But the difference is here we make use of locking techniques, where all the clients cannot access the shared file at once. So, whenever any client is connected to the server, it sends a request to the server. The corresponding request from the client is pushed into the queue.

Then, the server checks whether the shared file (critical section) is free, and grants access to the clients that ordered in the queue. After successfully accessing the file, client updates the value in the file by incrementing it. After updating, client sends a message to server to release the lock, and then the next request from the queue waits on the conditional variable till the other process releases the shared file i.e. releases the lock and signals on the conditional variable. Thus, a consistent locking mechanism is implemented.

Output:


ubuntuOS [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
Terminal
Nov 30 16:25
saiteja@saiteja-VirtualBox: ~/Desktop/Prjct2/dislock
Type here to search
54°F Cloudy
ENG
16:25
30-11-2022

```
saiteja@saiteja-VirtualBox:~/Desktop/Prjct2/dislock$ ./server 8010
How many number of clients to connect? 3
Connected to Client 4
Client 4 requesting access for the file.
Connected to Client 5
Client 5 requesting access for the file.
Connected to Client 6
Accessing........
Permission Granted to client: 4
Client 6 requesting access for the file.
Lock released by client: 4
Client 4 requesting access for the file.
Permission Granted to client: 5
Lock released by client: 5
Client 5 requesting access for the file.
Permission Granted to client: 6
Lock released by client: 6
Permission Granted to client: 4
Client 6 requesting access for the file.
Lock released by client: 4
Client 4 requesting access for the file.
Permission Granted to client: 5
Lock released by client: 5
Client 5 requesting access for the file.
Permission Granted to client: 6
Lock released by client: 6
Client 6 requesting access for the file.
Permission Granted to client: 4
Lock released by client: 4
Successfully file was accessed by client: 4
Permission Granted to client: 5
Lock released by client: 5
Client 5 requesting access for the file.
Permission Granted to client: 6
Lock released by client: 6
Client 6 requesting access for the file.
Permission Granted to client: 5
Lock released by client: 5
Permission Granted to client: 6
Successfully file was accessed by client: 5
Lock released by client: 6
Client 6 requesting access for the file.
Permission Granted to client: 6
Lock released by client: 6
Successfully file was accessed by client: 6
saiteja@saiteja-VirtualBox:~/Desktop/Prjct2/dislock$
```

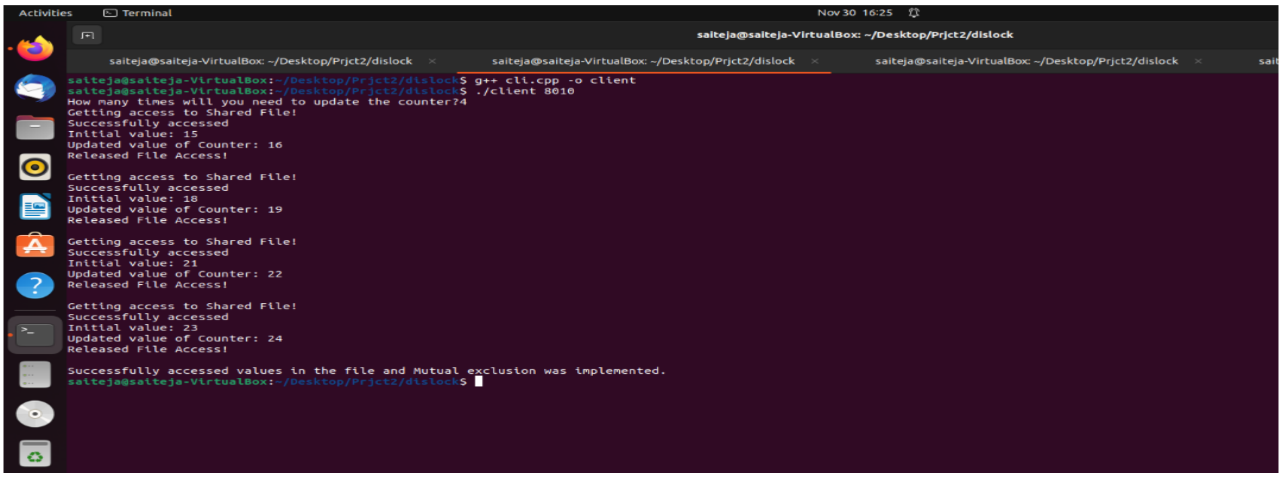

saiteja@saiteja-VirtualBox: ~/Desktop/Prjct2/dislock
saiteja@saiteja-VirtualBox: ~/Desktop/Prjct2/dislock
saiteja@saiteja-VirtualBox: ~/Desktop/Prjct2/dislock
saiteja@saiteja-VirtualBox: ~/Desktop/Prjct2/dislock

```
saiteja@saiteja-VirtualBox:~/Desktop/Prjct2/dislock$ g++ cli.cpp -o client
saiteja@saiteja-VirtualBox:~/Desktop/Prjct2/dislock$ ./client 8010
How many times will you need to update the counter?3
Getting access to Shared File!
Successfully accessed
Initial value: 14
Updated value of Counter: 15
Released File Access!

Getting access to Shared File!
Successfully accessed
Initial value: 17
Updated value of Counter: 18
Released File Access!

Getting access to Shared File!
Successfully accessed
Initial value: 20
Updated value of Counter: 21
Released File Access!

Successfully accessed values in the file and Mutual exclusion was implemented.
saiteja@saiteja-VirtualBox:~/Desktop/Prjct2/dislock$ 
```


Activities
Terminal
Nov 30 16:25
saiteja@saiteja-VirtualBox: ~/Desktop/Prjct2/dislock
saiteja@saiteja-VirtualBox: ~/Desktop/Prjct2/dislock
saiteja@saiteja-VirtualBox: ~/Desktop/Prjct2/dislock
saiteja@saiteja-VirtualBox: ~/Desktop/Prjct2/dislock

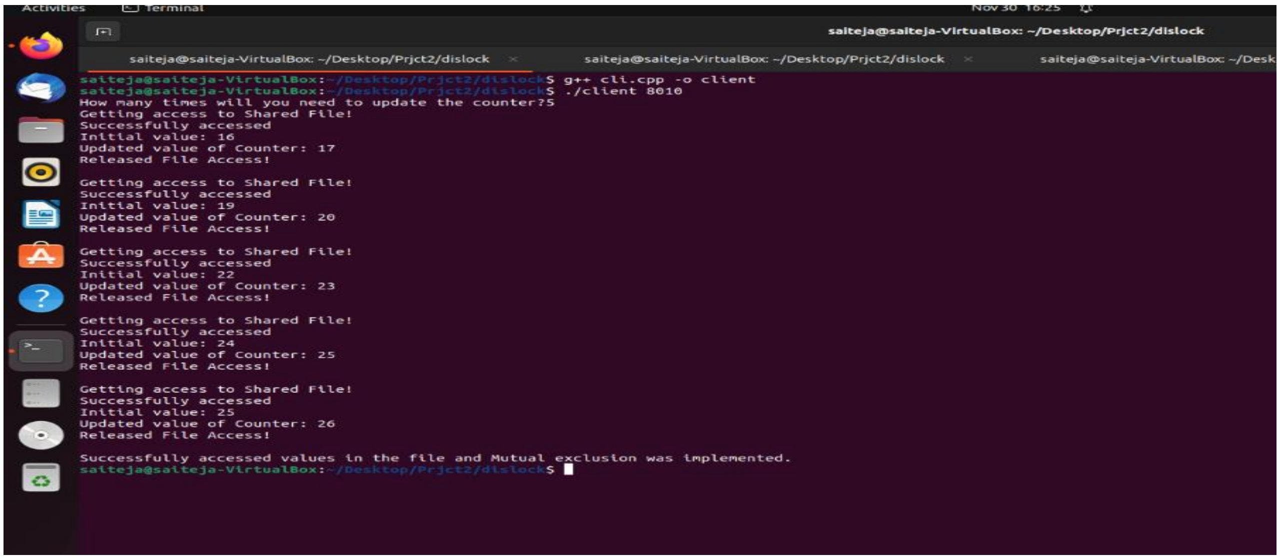
```
saiteja@saiteja-VirtualBox:~/Desktop/Prjct2/dislock$ g++ cli.cpp -o client
saiteja@saiteja-VirtualBox:~/Desktop/Prjct2/dislock$ ./client 8010
How many times will you need to update the counter?4
Getting access to Shared File!
Successfully accessed
Initial value: 15
Updated value of Counter: 16
Released File Access!

Getting access to Shared File!
Successfully accessed
Initial value: 18
Updated value of Counter: 19
Released File Access!

Getting access to Shared File!
Successfully accessed
Initial value: 21
Updated value of Counter: 22
Released File Access!

Getting access to Shared File!
Successfully accessed
Initial value: 23
Updated value of Counter: 24
Released File Access!

Successfully accessed values in the file and Mutual exclusion was implemented.
saiteja@saiteja-VirtualBox:~/Desktop/Prjct2/dislock$ 
```


Activities
Terminal
Nov 30 16:25
saiteja@saiteja-VirtualBox: ~/Desktop/Prjct2/dislock
saiteja@saiteja-VirtualBox: ~/Desktop/Prjct2/dislock
saiteja@saiteja-VirtualBox: ~/Desktop/Prjct2/dislock
saiteja@saiteja-VirtualBox: ~/Desk

```
saiteja@saiteja-VirtualBox:~/Desktop/Prjct2/dislock$ g++ cli.cpp -o client
saiteja@saiteja-VirtualBox:~/Desktop/Prjct2/dislock$ ./client 8010
How many times will you need to update the counter?5
Getting access to Shared File!
Successfully accessed
Initial value: 16
Updated value of Counter: 17
Released File Access!

Getting access to Shared File!
Successfully accessed
Initial value: 19
Updated value of Counter: 20
Released File Access!

Getting access to Shared File!
Successfully accessed
Initial value: 22
Updated value of Counter: 23
Released File Access!

Getting access to Shared File!
Successfully accessed
Initial value: 24
Updated value of Counter: 25
Released File Access!

Getting access to Shared File!
Successfully accessed
Initial value: 25
Updated value of Counter: 26
Released File Access!

Successfully accessed values in the file and Mutual exclusion was implemented.
saiteja@saiteja-VirtualBox:~/Desktop/Prjct2/dislock$ 
```

By implementing this, one can get to know how client-server architecture works with shared access using the locking mechanism.

While implementing this, we may observe that the requests need to wait until other requests are done. So, in order to make locking & unlocking atomically, we can make use of monitors (locks & condition variables).