PROJECT 2 REPORT

ABUL HASAN SAYEED – 1001911501

USAMAH MOIN MOHAMMED – 1002060937

I have neither given nor received unauthorized assistance on this work.

Signed: Abul Hasan Sayeed, Usamah Moin Mohammed Date: 10/10/2022

Vector clock implementation was done in python using sockets with one clock handler which checks the vectors for all three systems and updates them according to the order of called sending and receiving operations.

Text

Description automatically generated

After making sure that all the three systems are running in the background we could proceed sending and receiving messages. Here in this example, the first operation is done by sending a message from server 1 to server 2. The output of vector clock operations can be seen on individual systems.

The output that we receive on the clock script are the acknowledgements from the servers that we picked.

System1.py System2.py and System3.py are the scripts representing processes with unique ports connected on host ‘127.0.0.1’ within each of these three scripts there is a function definition called threaded process which creates a thread that handles listening to port and sending acknowledgements.

Order of operations:

1 -> 2, 1 -> 3, 3 -> 2

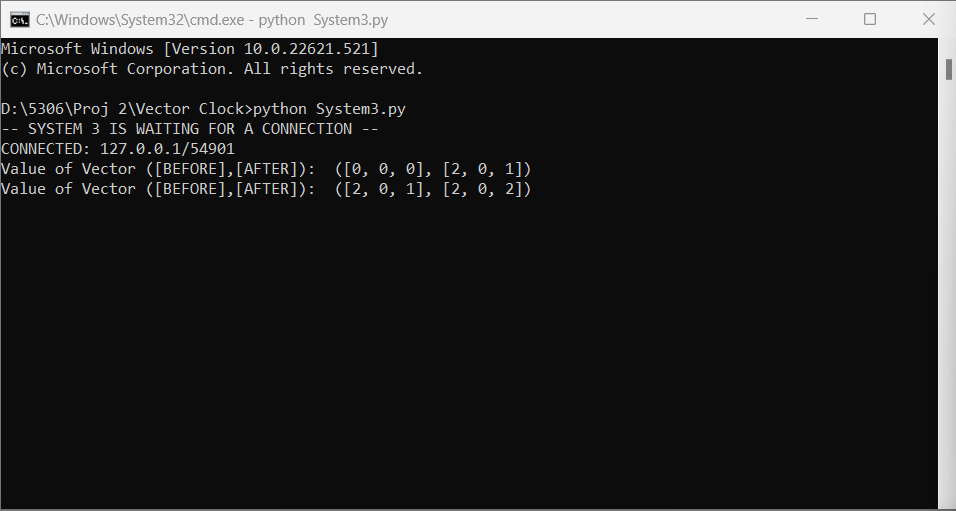
The output of vector clocks can be seen in the following images.

Text

Description automatically generated

Text

Description automatically generated



What we learned:

One of the biggest issues in distributed systems is to maintain a correct timeline of events and vector clock is used in distributed systems for providing consistent ordering of events through an artificial view of time.

In this project we have used sockets over RPC, Sockets are a way to enable inter process communication between the programs running. Using two threads one to listen through the communication port and one for sending messages to another node, we have created three severs that receive the message and send an acknowledgement back to the server as shown in the Clock image attached above.

Issues we encountered:

There are two issues we encountered, firstly was the reply of the messages i.e., before and after vector clocks which were to be sent were same in all the servers later, we realized that each node would update the clock of the other clock on their side only when the message is received. Secondly the list assignment operation “=”in python creates a reference pointer to the variable its assigned, using “.copy()” function we were able to create a copy of the before vector clock that is to be sent.