Logical Clocks - HW 2

Neil Patel

Experimental Report on Lamport Clock

The Lamport Clock serves the purpose of keeping track of which events occurs before other events. Essentially, when a message is sent from event A to event B, event B will be given a clock value higher than event A to indicate that the message will be received by event B after A has sent it. This idea is implemented on every event in a process with the clock value of a particular event being the maximum of the message's timestamp or its current clock value.

I implemented the Lamport Clock algorithm in Python using the Lamport.java code provided. The algorithm essentially, loops through the events of both processes, assigning a clock value to a particular event if the driver code indicates that the particular event received or sent a message to another event. The driver code has been altered to follow the time sequence provided for this assignment. This was done by changing the size of the two event lists to 7 and indicating that E12 sends a message to E23, E15 sends a message to E25 and that E17 receives a message from E24.

```
if __name__ == "__main__" :
    e1 = 7
    e2 = 7
    m = [[0]*7 for i in range(0,7)]

    m[1][2] = 1
    m[4][4] = 1
    m[6][3] = -1

    lamportLogicalClock(e1, e2, m)
```

The result of the code spits out the time sequence provided in the assignment:

```
neilpatel5412@Neil:/mnt/c/Users/neilp/Downloads$ python3 hw2.py
        e21     e22     e23     e24     e25     e26     e27
e11     0       0       0       0       0       0       0
e12     0       0       1       0       0       0       0
e13     0       0       0       0       0       0       0
e14     0       0       0       0       0       0       0
e15     0       0       0       0       1       0       0
e16     0       0       0       0       0       0       0
e17     0       0       0       -1      0       0       0
The time stamps of events in P1:
1 2 3 4 5 6 7
The time stamps of events in P2:
1 2 3 4 6 7 8
neilpatel5412@Neil:/mnt/c/Users/neilp/Downloads$
```