Name: Deep Walke
Roll Number: BT19CSE122

Distributed System Assignment 2

Output Screenshots

Question 1: Lamport's algorithm

```
58
59   const int numProcess = 5;
60   const int numIterations = 5;
61
62   void processFunction(int pid, LamportMutex& mutex) {
63       for (int i = 0; i < numIterations; ++i) {
64           mutex.lock(pid);
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
(base) deepwalke@Deeps-MacBook-Air A2 % g++ --std=gnu++17 Q1.cpp
(base) deepwalke@Deeps-MacBook-Air A2 % ./a.out
Process 0 locked the mutex, entering critical section
Process 0 unlocking the mutex, exited critical section
(base) deepwalke@Deeps-MacBook-Air A2 % g++ --std=gnu++17 Q1.cpp
(base) deepwalke@Deeps-MacBook-Air A2 % ./a.out
Process 0 locked the mutex, entering critical section
Process 0 unlocking the mutex, exited critical section
Process 1 locked the mutex, entering critical section
Process 1 unlocking the mutex, exited critical section
Process 0 locked the mutex, entering critical section
Process 0 unlocking the mutex, exited critical section
Process 1 locked the mutex, entering critical section
Process 1 unlocking the mutex, exited critical section
(base) deepwalke@Deeps-MacBook-Air A2 % g++ --std=gnu++17 Q1.cpp
(base) deepwalke@Deeps-MacBook-Air A2 % ./a.out
Process 0 locked the mutex, entering critical section
Process 0 unlocking the mutex, exited critical section
Process 1 locked the mutex, entering critical section
Process 1 unlocking the mutex, exited critical section
Process 2 locked the mutex, entering critical section
Process 2 unlocking the mutex, exited critical section
Process 3 locked the mutex, entering critical section
Process 3 unlocking the mutex, exited critical section
Process 4 locked the mutex, entering critical section
Process 4 unlocking the mutex, exited critical section
Process 0 locked the mutex, entering critical section
Process 0 unlocking the mutex, exited critical section
Process 1 locked the mutex, entering critical section
Process 1 unlocking the mutex, exited critical section
Process 2 locked the mutex, entering critical section
Process 2 unlocking the mutex, exited critical section
Process 3 locked the mutex, entering critical section
Process 3 unlocking the mutex, exited critical section
Process 4 locked the mutex, entering critical section
Process 4 unlocking the mutex, exited critical section
Process 0 locked the mutex, entering critical section
Process 0 unlocking the mutex, exited critical section
Process 1 locked the mutex, entering critical section
Process 1 unlocking the mutex, exited critical section
Process 2 locked the mutex, entering critical section
Process 2 unlocking the mutex, exited critical section
Process 3 locked the mutex, entering critical section
Process 3 unlocking the mutex, exited critical section
Process 4 locked the mutex, entering critical section
```

1 process
1 iterations

2 process
2 iterations

5 process
5 iterations

Question 2: Lamports Logical clock.

```
(base) deepwalke@Deeps-MacBook-Air A2 % g++ --std=gnu++17 Q2.cpp
(base) deepwalke@Deeps-MacBook-Air A2 % ./a.out 2
Enter the sender and receiver process id: 0 0
Enter the next event number for this process id:1
The logical clock for each process is:
Process 0: 0 1
Process 1:

Enter the sender and receiver process id: 1 1
Enter the next event number for this process id:1
The logical clock for each process is:
Process 0: 0 1
Process 1: 0 1

Enter the sender and receiver process id: 0 0
Enter the next event number for this process id:2
The logical clock for each process is:
Process 0: 0 1 2
Process 1: 0 1

Enter the sender and receiver process id: 0 1
Enter the event number of sender and reciever: 2 2
The logical clock for each process is:
Process 0: 0 1 2
Process 1: 0 1 3

Enter the sender and receiver process id: 0 0
Enter the next event number for this process id:3
The logical clock for each process is:
Process 0: 0 1 2 3
Process 1: 0 1 3

Enter the sender and receiver process id: 0 1
Enter the event number of sender and reciever: 3 3
The logical clock for each process is:
Process 0: 0 1 2 3
Process 1: 0 1 3 4

Enter the sender and receiver process id: 0 0
Enter the next event number for this process id:4
The logical clock for each process is:
Process 0: 0 1 2 3 4
Process 1: 0 1 3 4

Enter the sender and receiver process id: 1 1
Enter the next event number for this process id:4
The logical clock for each process is:
Process 0: 0 1 2 3 4
Process 1: 0 1 3 4 5
```

Question 3:
“Chandy Misra Haas” (CMH) Edge chasing algorithm.