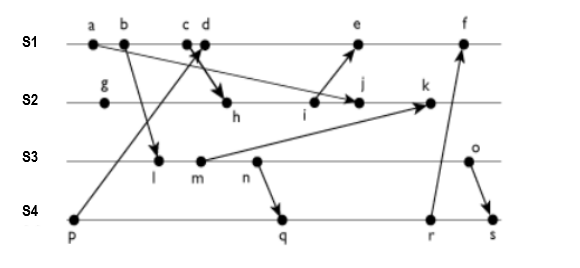
|  |  |  |  |
| --- | --- | --- | --- |
| **paf_kiet_logo** | **COLLEGE OF COMPUTING AND INFORMATION SCIENCES** | | |
| **Assignment 3 (Summer 2021 Semester)** | | |
| **Class Id** | 107945,6,7,8 | **Course Title** | P&DC |
| **Due Date** | 26-12-21 | **Total Marks** | 10 |
| **Student Id** |  | **Student Name** |  |

**Instructions:**

* Fill out your Student ID and Student Name in above header.
* Do not remove or change any part question paper.
* Write down your answers with title “Answer for Question# 00”.
* In case of CHEATING, COPIED material or any unfair means would result in negative marking or ZERO.
* Submit this file on LMS
* Late submissions are not accepted at all.
* Save file with name: [StudentID \_StudentName\_ClassID]

Q1. Lamport Logical Clocks algorithm was a remarkable addition in distributed systems by Leslie Lamport.

Suppose there are 4 servers, each server have multiple send, received and instruction happened events. All servers started with the zero (0) logical clock but later on clocks have moved to different integers. Clock rate will be same for all clock which is 1.

**Your student ID integers are now the starting clocks for servers respectively (Suppose your student ID is 57616, the starting clock for server1 (S1), server2 (S2), server3 (S3), and server4 (S4) are 5, 7, 6, and 1 respectively.**

Assuming that the system implements Lamport’s timestamps draw a picture illustrating the

timestamps or the messages and explain how the timestamps are obtained.

Q2. There are 5 computers: the daemon and the clients A, B, C, and D. The daemon supplies time should be the highest digit in your student id and the other 4 digits will become hours for clients and take minutes for client according to your date of birth **(Suppose your id is 57616 and your DOB is 04/08/1994, the clock time of daemon will become 7:00, for Client A: 5:04, Client B: 6:08, Client C: 1:19 and Client D: 6:94 after minute adjustment will be 7:34)**. Which time would all the clocks show after synchronization using Berkley’s Algorithm?

**(If any student is having less than 5 digit SID he/she can use one random digit and if number is greater than 60 for minute then adjust it.**

Q3. Write an MPI program to compute to send array from one process and find its sum in other process. Root process will be ranked zero where declaration and initialization of arrays is done and send it to process 5. Now Process 3 will perform the sum operation and returns the sum to the root process.