Assignment 3

Written problems (Individual)

Problem 1
For each of the systems of distributed processes shown below, compute the following:

1. Scalar times of all events using Lamport algorithm
2. Vector times of all events

1)

2)
Problem 2

In your own words, solve exercise 3.3 (page 84) in the Kshemkalyani and Singhal text book.

Problem 3

Solve exercise 4.3 (page 122) in the Kshemkalyani and Singhal text book.
Programming problem (Pairs)
For this problem, you must use Amazon EC2 instances as the independent processes. I recommend you use Java RMI, but you are free to use any programming language you want.

Setting up your account
Create your Amazon AWS account and create your first distributed computation. You can use any of the available AWS tutorials, or you can use this simplified tutorial that I created for you.

Global Snapshot
In this problem, you will implement a virtual currency distributed system. Each process in the system represents a single account, and processes can transfer money to each other through message passing.

For example, in the system shown below, we have two processes with initial balances of $600 and $200 respectively, and a total of $800 in the system. S1 decides to transfer $50 to S2, and S2 later transfers $80 to S1. Eventually, S1 and S2 have balances of $630 and $170 respectively, and a total of $800.

Your goal is to implement the Chandy-Lamport algorithm to obtain snapshots of the system.
Implementation details

Logic of each independent process
Each account should have an initial balance of $200, and that balance will increase and decrease with every transfer.

In the lifetime of the process (i), it will act as follows:

1. Let $R$ = a random number between 5000 and 50000
2. Let $M$ = a random number between 1 and local balance of i
3. Let $P$ = a random number between 1 and 4, not including i
4. After $R$ milliseconds, transfer $M$ to process $P$

Upon receiving a transfer, the process will:

1. Increase the local balance with the amount attached in the transfer

Note that we assume that each process knows the address of every other process in the system.

Leader election
To get a snapshot, one of the processes needs to initiate the process, and that process is picked using a leader election algorithm as that discussed in lecture.

To simplify this program, design in a way such that the leader is picked only once at the beginning of the system. Transfer of money occurs after a leader is determined.

Get snapshot
Once the leader is decided, initiate the process of taking a global snapshot. Use the Chadny-Lamport algorithm to get this snapshot