

**CS 4445/9644: Special Topics on Scalable and Robust Distributed Systems**  
**Department of Computer Science**  
**The University of Western Ontario**  
**Programming Assignment 2**  
**Spring, 2022**

**Purpose**

Distributed computing systems are complex and difficult to understand. This assignment will give you a better understanding of the challenges in implementing distributed algorithms by using Lamport clocks to totally order operations at each replica. You will also learn the fundamentals of developing a peer-to-peer application where a process is both a client and a server.

**Platform**

To support this assignment we have three VMs available for your use as follows:

- cs4435.gaul.csd.uwo.ca
- cs4435-vm2.gaul.csd.uwo.ca
- cs4435-vm3.gaul.csd.uwo.ca

**Terminology**

In Assignment 1 there was a clear distinction between the client and the server. In this assignment each process is both a client and a server. We will refer to this as a node.

**Assignment Description**

You are to develop an application that uses Lamport Logical Clocks to totally order operations on N replicas. Each node hosts a replica. Assume that the replicas represent bank accounts. Each node can initiate an operation on the bank account. You are to use Lamport Logical Clocks to ensure that each node uses the same order of operations.

**Input**

- Account Information: All nodes read the same file. This will be accounts.json
- Command line arguments: port number, name of file with operations, name of node.
  - Example node names: Node 1, Node 2... or 1, 2, 3
  - A file with operations should have at the end the word *done*. Please note that there is a file for each node.

**Output**

- Each node outputs an output file once all nodes are done. If the node's name is "Node 1" then its output file should be accountsUpdatedNode1.json. If the node's name is 1 then its output file should be accountsUpdated1.json.

**Running the Code**

Compiling is similar to what you had to do for Assignment 1.

Let's say that you run each instance of a node on the same machine. This means that the port numbers need to be different e.g.,

```
go run node/main.go "Node 1" :5000 localhost:8500  
go run node/main.go "Node 2" :5001 localhost:8500
```

The service discovery program uses port 8500. The two nodes use ports 5000 and 5001.

### **Directory Organization**

The directory with the code should be lamportClocks. Within this you should have the following directories:

- proto: This has the proto file
- node: This has the code
- bin: This will have the executables and all input/output files.

### **Sample Code**

Sample code will be given to you through OWL that highlights how to do a peer-to-peer application. It is based on the hello service example code provided for assignment 1. The code is from the company that provides the service discovery protocol being used for this assignment.