Lamport’s Logical Clocks

Project 1

CPSC 474

19 September 2020

Dr. Doina Bein

David Nguyen 891362089

**Summary**

**Pseudocode**

class Process{

private:

vector<string> configuration;

public:

Process(config){

this->configuration = config;

}

void send(){

this->configuration.push\_back(“S” + SEND\_ID++ );

}

void receive(){

this->configuration.push\_back(“R” + RECEIVE\_ID++);

}

};

//const NUM\_PROC = 5;

// at most 9 send events

static int SEND\_ID = 1;

static int RECEIVE\_ID = 1;

vector<Process> sample1{};

**Calculate**

vector<vector<int>> calculateLC(vector<Process> events){

vector<vector<int>> result;

vector<string> recieve;

int lc = 1;

for(int i = 0;

//If *a* is the first event and is an internal or send event, then LC(*a*) = 1

if(events)

. //If *a* is the first event and is a receive event, then LC(*a*) = *k* + 1 where *k* is the LC-value of the send event corresponding to *a* (that has occurred at a process other than P).

else if()

//If *a* is not the first event and is an internal or send event, then LC(*a*) = *k* + 1 where *k* is the LC-value of the event just before *a* at process P.

else if()

//If *a* is not the first event and is a receive event, let *b* be the send event corresponding to *a* (that has occurred at a process other than P) and *k* be the clock value of the event just before *a* at process P. Then LC(a) = max{ k, LC(b) } + 1

else if()  
  
 return;

}

**Verify**

**Screenshots**