Lamport Clock

Hou, Jue

[jue.hou@helsinki.fi](mailto:jue.hou@helsinki.fi)

* Solution

1. Keep the main process stay in the state of monitoring the port and waiting for new messages.
2. Start a thread especially for executing random event
3. Once a message comes in, node will start a new thread and assign the thread to handling the message according to the lamport clock algorithm.
4. Once a new clock message from other node arrived, the node will start a new thread and assign the thread for handling the message and update the local clock according to the lamport clock algorithm. After sending the clock message successfully, sender’s clock increase by 1.
5. No random event is allowed to execute after the message is under processing until the handling thread has ended. There hence will be almost nothing related to concurrency.
6. Since the exercise file does not mention anything about how many nodes are there, how many node will there be and whether nodes know each other, there is an initiating process for every node to test the connection between each other. There hence have to be a report for initialization, or it might lead to a wrong result.

* Executing Process

1. Got started by user manually
2. Node will check if the configuration is valid.
   1. If the configuration does not exist in the file or the port is already in use, the node will report fail and terminate automatically.
   2. If the configuration is valid, node will record its information in an internal list. I name it as “LIST”
3. Check for the existence of nodes. The current node will send a confirmation message including its own id and port to all the nodes according to the configuration file.
   1. If message is delivered successfully, record the id and port of the receiver into LIST
   2. If message cannot be delivered, continue to next node.
4. After all the nodes have been checked, node will turn to a continuous state to wait for connection and accept it. There will be three different kinds of messages.
   1. Confirmation message: Node will record the id and port of the sender into LIST
   2. Starting message: Node will start the whole simulation, which also means start the random event thread. This message will be send by running “lamport-master.py”
   3. Clock message: Start a new thread and assign it to handling the message. According to the lamport algorithm, the timestamp in the message will be compared with the local clock. The greater one will be plus one and served as the new local clock. After that, output line as requested.
5. Start the simulation manually by running “lamport-master.py”. Node will also report other relevant information. **But that is not output or a part of output.**
6. Random event thread started. Two events will be randomly select one to be executed. Event will be executed once every 0.5 second. All the random factor will be generated by “randrange()” function and be converted to the suitable form. After every event, node will report it in the screen. Although report is in the same format of output, **this is not output either.**
   1. If there is no clock message handling thread, loop to execute random events for 100 times.
   2. If there is at least one clock message handling thread, loop with no increment on counter and do nothing until the thread ended.
7. After 100 time random events, random event thread will sleep for 2 second in case some clock messages are still on their way.
8. **Output into a text file.** Output file is named in the format as “ouput-id-port.txt”. For example, if the id of a node is 1 and its port is 10001, than the output file for this node is “output-1-10001.txt”
9. By using “os\_exit(0)”, random event thread will terminate the whole node.

* Instruction

1. Environment Requirement and Programming Language

Python 3+ with correct configuration

1. Running procedure
2. Start a new command line interface and enter the folder of the program
3. Run “lamport.py” followed by the path of configuration file and the line for the configuring of the node in the interface. Example show as follow:



In this case, “temp.txt” is my configuration file and line for configuring is the first line of the file. “temp.txt” is now in the same folder of “lamport.py”.

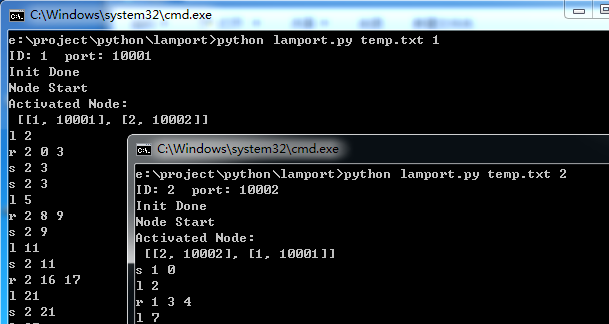
1. Node will report its own id and port. Wait until the node shows “Init Done”, repeat the procedure 1 and 2 to start another node.



1. If user believes all the nodes have been activated and the last node also has reported “Init Done”, start a new command line interface and run the “lamport-master.py” followed by the path of configuration file. Example shows as follow:



1. All the activated will report the list of activated node and start the simulation.



1. Check the output file.