Homework 6 - Chandy Lamport Algorithm to capture a Distributed Snapshot

Overview:

- Simulation has been implemented on 3 processors P1, P2 and P3, which send MARKER, ALGORITHM, SEND, RECEIVE and COMPUTATION messages to each other.
- Each processor has its own execution plan, incoming and outgoing channel.
- Processor P1 initiates snapshot and records its own state. It sends MARKER messages to Processor P2 and P3.
- Any processor on receiving a first MARKER message initiates a Thread Recorder, which starts monitoring messages for its corresponding incoming channel. Thread Recorder stops recording when it receives a duplicate marker.
- When all the processors receive a duplicate marker the algorithm terminates.

Classes:

- Buffer.java
- Main.java
- Message.java
- MessageType.java
- Processor.java
- ThreadRecorder.java

Input:

Execution plan for 3 processors P1, P2, P3.

Output:

Distributed Snapshot of the system, which gives information regarding number and type of messages recorded on each channel after receiving first Marker message until a duplicate marker.

(Note: Output.txt for program output has been uploaded to github. Output has been recorded using java.util.Logger since it is thread safe. The use of logger prevents overlapping of outputs from different threads.)