

8 Concurrent and Distributed Systems (mk428)

- (a) State the definition of the *happens-before* relation in a distributed system. [2 marks]
- (b) Describe how to compute Lamport timestamps. [2 marks]
- (c) Lamport timestamps are often used to provide a total ordering on events in a system with multiple communicating processes. State how this total order is defined, and explain how this order relates to the happens-before relation. [2 marks]
- (d) Explain what is meant by *FIFO broadcast* and *FIFO-total order broadcast* in the context of a process group. [2 marks]
- (e) Give pseudocode for an algorithm that implements FIFO-total order broadcast using Lamport clocks. You may assume that each process has a unique identifier, and that the set of all process IDs in the group is known. Further assume that the underlying network provides reliable FIFO broadcast; that is, you may use a function `sendFIFO(m)` that transmits message `m` to all processes in the group. The function `deliverFIFO(m)` is invoked at each process (including the sender) when message `m` is delivered at that process by the FIFO broadcast layer. Your algorithm should call `deliverTotalOrder(m)` with messages `m` in strictly monotonically increasing order of their Lamport timestamp.

Use this outline to get started:

```
// Called by the user when they want to send a message
function totalOrderBroadcast(msg) {
    let m = ...; // construct message to send by FIFO broadcast
    sendFIFO(m); // use underlying FIFO broadcast
}

// Called by the FIFO broadcast layer when a message is received
function deliverFIFO(m) {
    // Figure out the total order...
    // then call deliverTotalOrder(...) as appropriate
}
```

[10 marks]

- (f) Briefly comment on the fault-tolerance properties of your algorithm in Part (e). [2 marks]