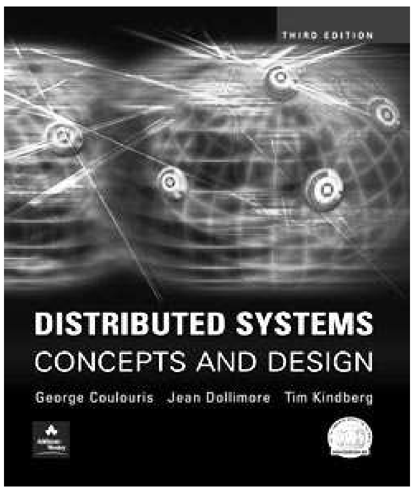
# Exercises for Chapter <10:

## TIME AND GLOBAL STATES




*From* **Coulouris, Dollimore and Kindberg**

**Distributed Systems: Concepts and Design**

# Exercise 10.1

- Why is computer clock synchronization necessary? Describe the design requirements for a system to synchronize the clocks in a distributed system.

*page 386*

# Exercise 10.2

⌘ A clock is reading 10:27:54.0 (hr:min:sec) when it is discovered to be 4 seconds fast. Explain why it is undesirable to set it back to the right time at that point and show (numerically) how it should be adjusted so as to be correct after 8 seconds has elapsed.

*page 390*

# Exercise 10.3

⌘ A scheme for implementing at-most-once reliable message delivery uses synchronized clocks to reject duplicate messages. Processes place their local clock value (a ‘timestamp’) in the messages they send. Each receiver keeps a table giving, for each sending process, the largest message timestamp it has seen. Assume that clocks are synchronized to within 100 ms, and that messages can arrive at most 50 ms after transmission.

(i) When may a process ignore a message bearing a timestamp *T*, if it has recorded the last message received from that process as having timestamp *T'*?

(ii) When may a receiver remove a timestamp 175,000 (ms) from its table? (Hint: use the receiver’s local clock value.)

(iii) Should the clocks be internally synchronized or externally synchronized?

⌘ *page 391*

# Exercise 10.4

- A client attempts to synchronize with a time server. It records the round-trip times and timestamps returned by the server in the table below.

  Which of these times should it use to set its clock? To what time should it set it? Estimate the accuracy of the setting with respect to the server's clock. If it is known that the time between sending and receiving a message in the system concerned is at least 8 ms, do your answers change?

| *Round-trip (ms)* | *Time (hr:min:sec)* |
|---|---|
| 22 | 10:54:23.674 |
| 25 | 10:54:25.450 |
| 20 | 10:54:28.342 |

*page 391*

# Exercise 10.5

- In the system of Exercise 10.4 it is required to synchronize a file server's clock to within ±1 millisecond. Discuss this in relation to Cristian's algorithm.

*page 391*

# Exercise 10.6

⌘ What reconfigurations would you expect to occur in the NTP synchronization subnet?

*page 394*

# Exercise 10.7

- An NTP server B receives server A's message at 16:34:23.480 bearing a timestamp 16:34:13.430 and replies to it. A receives the message at 16:34:15.725, bearing B's timestamp 16:34:25.7. Estimate the offset between B and A and the accuracy of the estimate.

*page 395*

# Exercise 10.8

- Discuss the factors to be taken into account when deciding to which NTP server a client should synchronize its clock.

*page 396*

# Exercise 10.9

- ⌘ Discuss how it is possible to compensate for clock drift between synchronization points by observing the drift rate over time. Discuss any limitations to your method

*page 397*

# Exercise 10.10

- ⌘ By considering a chain of zero or more messages connecting events *e* and *e'* and using induction, show that $e \rightarrow e' \Rightarrow L(e) < L(e')$.

*page 398*

# Exercise 10.11

- Show that $V_j[i] \leq V_i[i]$.

*page 399*

# Exercise 10.12

- In a similar fashion to Exercise 10.10, show that $e \rightarrow e' \Rightarrow V(e) < V(e')$.

*page 400*

# Exercise 10.13

- Using the result of Exercise 10.11, show that if events *e* and *e'* are concurrent then neither $V(e) \leq V(e')$ nor $V(e') \leq V(e)$.
- Hence show that if $V(e) < V(e')$ then $e \rightarrow e'$

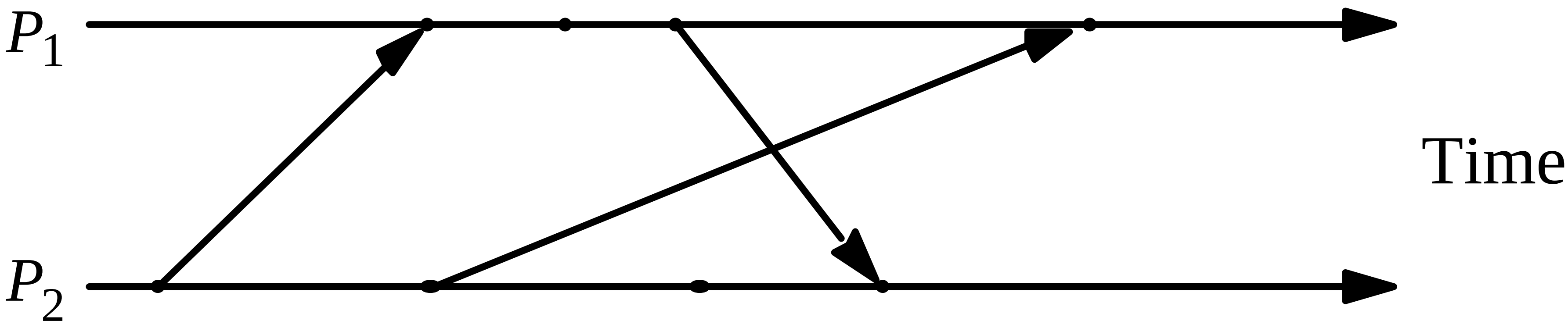
*page 400*

# Exercise 10.14

- Two processes *P* and *Q* are connected in a ring using two channels, and they constantly rotate a message *m*. At any one time, there is only one copy of *m* in the system. Each process's state consists of the number of times it has received *m*, and *P* sends *m* first. At a certain point, *P* has the message and its state is 101. Immediately after sending *m*, *P* initiates the snapshot algorithm. Explain the operation of the algorithm in this case, giving the possible global state(s) reported by it.

*page 405*

# Exercise 10.15




⌘ The figure above shows events occurring for each of two processes, *p*1 and *p*2. Arrows between processes denote message transmission.

Draw and label the lattice of consistent states (*p*1 state, *p*2 state), beginning with the initial state (0,0).

*page 412*

# Exercise 10.16

- Jones is running a collection of processes $p_1, p_2, \ldots, p_N$ . Each process $p_i$ contains a variable $v_i$. She wishes to determine whether all the variables $v_1, v_2, \ldots, v_N$ were ever equal in the course of the execution.

  (i) Jones' processes run in a synchronous system. She uses a monitor process to determine whether the variables were ever equal. When should the application processes communicate with the monitor process, and what should their messages contain?

  (ii) Explain the statement *possibly* ($v_1 = v_2, = \ldots = v_N$ ). How can Jones determine whether this statement is true of her execution?

*page 413*