22C:166: Distributed Systems and Algorithms

Homework 5

Assigned on 11/29/05 due 12/05/05 11:59:59 PM. Please submit a typewritten solution and email it to your TA by the deadline.

Total points = 50 points

Question 1. Consider two read write object X, Y, and the history of read and write operations by two processes P0, P1 as shown in Fig. 17.10:

\[
\begin{align*}
W & (X:= 1) \quad W & (X:=2) \quad R & (Y:= 1) \\
\hline
\hline
P0 & \hline
\hline
Y:=1 \quad Y:=2 \quad R & (X = 2) \quad R & (Y:=2)
\end{align*}
\]

Fig 17.10. A history of read and writes by two processes

Assume that the timelines are drawn to scale. Determine if the above history reflects linearizability, or sequential consistency, or none of the above. Briefly justify.

Question 2. A quorum-based replica management system engages only a designated fraction of the replicas for every read and write. Here is a description of how it works:

Let there be \( N \) servers. To complete a write, a client must \textit{successfully} place the updated data item on \( W \ (1 \leq W \leq N) \) servers. The updated data item will be assigned a new version number generated by incrementing its current version number. To read the data, the client must read out the copies from any set of at least \( R \ (1 \leq R \leq N) \) servers. Of these, it will find out a copy with the highest version number, and accept that copy.

Assume that \( N \) is an even number.

\begin{enumerate}
\item Will sequential consistency be satisfied if \( W = N/2, R = N/2 \)? \textbf{Briefly justify.}
\item Will sequential consistency be satisfied if \( W = (N/2)+1, R = (N/2)+1 \)? \textbf{Briefly justify}
\end{enumerate}
(c) If \( N = 10, W = 7 \), then what should be the \textit{minimum} value of \( R \) so that sequential consistency is satisfied? \textbf{Briefly justify.}

\textbf{Question 3.} Many processors use a \textit{write buffer} to speed up the operation of its write-through cache memory. It works as follows: When a variable \( x \) is updated, the processor writes its value into the local cache \( C \), and at the same time puts the updated value into a \textit{write buffer} \( W \). A separate controller then transfers this value into the main memory. The advantage is that the processor does not have wait for the completion of the write memory operation, which is slow. This speeds up instruction execution. For a read operation, data is retrieved from the local cache.

Consider running the following program executed by two concurrent processes \( P1, P2 \) on the above multiprocessor. Here, \( x, y \) are two shared variables

\begin{align*}
\text{Process P1} & \quad \text{Process P2} \\
\{\text{initially } x = 0\} & \quad \{\text{initially } y = 0\} \\
x := 1; & \quad y := 1; \\
\text{if } y = 0 \text{ \{ } x := 2 \text{ \}}; & \quad \text{if } x = 0 \text{ \{ } y := 2 \text{ \}}; \\
\text{Print } x & \quad \text{Print } y
\end{align*}
Will sequential consistency be satisfied? Explain your answer.

**Question 4.** The table below gives ordered lists of send and receive events of six multicast messages on four processors in a group communication system. Let \( S(M_j,i) \) represent the event of sending message \( M_j \) at time \( i \), and \( R(M_j,k) \) represent the event of receiving message \( M_j \) at time \( k \). Assume that a processor receives its own message immediately.

Processor 1: \( S(M_1,0), R(M_3,5), R(M_2,8), R(M_4,12), R(M_6,13), R(M_5,14) \)
Processor 2: \( R(M_1,2), S(M_2,3), S(M_3,4), R(M_4,14), R(M_6,15), R(M_5,16) \)
Processor 3: \( R(M_3,6), R(M_1,7), S(M_6,8), S(M_5,9), R(M_2,11), R(M_4,12) \)
Processor 4: \( R(M_3,6), R(M_1,8), R(M_5,10), S(M_4,11), R(M_2,12), R(M_6,13) \)

The group communication system implements causal order multicast semantics.

Give the vector clock values associated with the transmission, reception and delivery of each message. Also, show the local clock value(s) at which each message first becomes eligible for delivery on each processor.

**Question 5.** Consider a board game being played on the network by a group of players. The board consists of nine squares numbered 0 through 8. Each user’s move consists of clicking on two squares that will swap the positions of those two squares on the board. The outcome of the game is not relevant here.

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
</tbody>
</table>

To play on the network, each user has a replica of the game board in front of her on the screen. The players send messages declaring their moves to this group. A legal
configuration of the board is one that is reachable from the initial configuration by a sequence of moves, and due to possible concurrent moves by the players, the board should never move to an illegal configuration.

What is the weakest type of ordered multicast that must be used in this game? Explain your answer.