Review for 166 Exam 3

Group communication
Various types of multicasts, their significance and implementation
Open groups and View-synchronous communication

Replication
Passive vs active replication
Fault-tolerant state machine
Consistency models

Self-stabilization
Understanding the concept
Convergence and closure properties

Question 1. Four replica servers S₁ through S₄ are required to coordinate amongst themselves to create the illusion of a single highly reliable server S that will service a set of clients. One of the servers can potentially undergo byzantine failure, but no one knows which one. To achieve fault-tolerance, updates received by a server will be multicast to all servers in the system

(a) What kind of multicast will be used to propagate the updates?
(b) How will the byzantine failure of a server be handled?

Question 2. The members of a group use view-synchronous communication to communicate with one another. Initially, there are four processes 0, 1, 2, 3. Process 0 sent a message m to the group in view (0,1,2,3). Processes 0, 1 and 2 delivered message m in the same view, but process 3 did not. Is this an acceptable behavior? Justify your answer.

Question 3. Four processes A, B, C, D execute the following actions on two shared variables x and y:

A
\[ \text{W(x:=10)} \]

B
\[ \text{W(x:=20)} \]

C
\[ \text{R(x=10)} \]
\[ \text{W(y:=5)} \]
\[ \text{R(x=20)} \]

D
\[ \text{R(x=20)} \]
\[ \text{R(x=10)} \]
\[ \text{R(y=5)} \]

(a) The above trace satisfies sequential consistency but not causal consistency.
(b) The above trace satisfies both sequential and causal consistency.
(c) The above trace satisfies causal consistency but not sequential consistency.
(d) The above trace satisfies neither sequential nor causal consistency.

**Question 4.** A quorum-based replica management system works as follows: Let there be \( N \) servers. To complete a write, a client must *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. If \( N = 10 \), \( W = 7 \), then what should be the *minimum* value of \( R \) so that sequential consistency is satisfied?

**Question 5.** Many shared memory multiprocessors use a *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 its *write buffer* \( W \). A separate controller then transfers this value into the main memory and invalidates all other cache entries storing \( x \). The advantage is that the processor does not have to wait for the completion of the write memory operation, which is slower than writing into the buffer. This speeds up instruction execution. For a read operation, data is retrieved from the local cache. However, if the cache entry is invalid, then data is retrieved from the main memory.

Consider the following programs executed by \( P_1 \) and \( P_2 \)

\[
\begin{align*}
P_1: & \text{ Initially } x = 0; \quad x := 1; \quad \text{print } y \\
P_2: & \text{ Initially } y = 0; \quad y := 1; \quad \text{print } x
\end{align*}
\]

(a) If sequential consistency is maintained, then what value(s) of \((x, y)\) will *never be* printed?
(b) What values of \((x, y)\) may be printed by the shared-memory multiprocessor of Figure 1?