
Question 1
Give an example to show that if transactions do not follow two-phase locking protocol, then it is possible to have non-serializable schedules.
For example, give a schedule with 2 transactions that do not follow two-phase locking, such that the schedule is not serializable.
Make the example as simple as you possibly can. Justify your answer.

Question 2
Give an example of a schedule with 3 transactions T1, T1, and T3 to demonstrate cascading aborts. Transactions should follow two-phase locking protocol (include the lock and unlock instructions). Aborting T1 (for any reason) should cause T2 to be forced to abort, and aborting T2 should cause T3 to be forced to abort. Justify your answer.

Question 3
Give an example of a schedule with 3 transactions T1, T1, and T3 to demonstrate deadlock (that involves 3 transactions). Justify your answer.

Question 4
Consider a deadlock detection and recovery system that selects the transaction to be aborted (the victim) according to the following rule:
Transaction with largest number of remaining instructions is selected.
Is there a possibility of starvation? Why (justify your answer).
Answer the same question if the following rule is used instead:
Transaction with smallest number of remaining instructions is selected.

Question 5
Consider the following two transactions

\[
\begin{align*}
T1 & \quad \quad \quad T2 \\
\text{read}(A) & \quad \quad \quad \text{read}(B) \\
\text{read}(B) & \quad \quad \quad \text{read}(A) \\
\text{if } A=0 \text{ then } B=B+1 & \quad \quad \quad \text{if } B=0 \text{ then } A=A+1 \\
\text{write}(B) & \quad \quad \quad \text{write}(A)
\end{align*}
\]

(a) Add lock, upgrade, and unlock instructions so that they observe the two-phase locking (with lock conversions) protocol. Write down the transactions.
(b) Can the execution of these transactions result in a deadlock. Explain how.