



Assignment 4

Due date: March 19, 2004 **Due in class**

1. Consider the following transactions:

T1: R(A), W(A); R(B), Commit
T2: R(A), R(B), W(B), Commit
T3: R(C); R(A), R(B), W(C), Commit

In this question you have to find schedules for these transaction according to different protocols. The schedules should include not only the read and write operations of the transactions, but also the actions required by the respective protocol (e.g. “request Xlock(C), granted”, “request SLock(A), denied”, “Wait”, “Abort/Rollback”, or “request lockUpgrade”).

The transactions should be scheduled in a *round-robin fashion*, i.e., first, transaction T1 is allowed to execute its first action, then T2, then T3, and after that again T1, and so on – if that is possible according to the protocol (a transaction may have to wait, in which case the next transaction can try to execute its next action). In addition: if a transaction is aborted, it should be *restarted* again after the other transactions have executed their next operation (if possible).

- a. Describe in detail how Strict 2PL with the “wound-wait” as well as the “wait-die” deadlock resolution mechanisms would schedule the given transactions. You should assume that a transaction always requests the lowest possible lock that it needs for the current operation – if it does not already hold a lock that allows the current operation. If it already holds a lock on the object, but that lock does not allow the current operation, the transaction will request a lock upgrade. If a transactions gets a requested lock, it can execute the operation for which it needed the lock immediately (possibly after another transaction may have been forced to abort).
 - b. Generate a schedule for the above transactions according to the Timestamp Ordering protocol. Include columns for the timestamps of objects and transactions when you write up your schedule.
2. State which of the following schedules are serializable. Justify your answer by giving the serialization graph of each schedule. If a schedule is serializable, give its equivalent serial schedule.
 - a. r1(x) r2(y) r1(z) r3(z) r2(x) r1(y)
 - b. w1(x) r2(y) r1(z) r3(z) r1(x) w2(y)
 - c. r1(z) w2(x) r2(z) r2(y) w1(x) w3(z) w1(y) r3(x)
 - d. r1(x) w2(y) r1(y) r3(z) w2(z) w4(y) w3(x)