Fall 2004 Exercise Questions
Week 14 – Ending 12/03/04

1. In general, is it possible to have a deadlock when the regular two-phase locking (i.e., non-strict) protocol is obeyed? If yes, give an example; if not, explain briefly. What happens with strict 2PL?

2. Consider the following two transactions:
   T1 = w1(C) r1(A) w1(A) r1(B) w1(B);
   T2 = r2(B) w2(B) r2(A) w2(A)
   Say our scheduler performs exclusive locking only (i.e., no shared locks). For each of the following three instances of these transactions annotated with lock and unlock actions, say whether the annotated transactions:

   1.obey two-phase locking,
   2.will necessarily result in a conflict serializable schedule (if no deadlock occurs),
   3.will necessarily result in a schedule that avoids cascading rollback (if no deadlock occurs),
   4.will necessarily result in a strict schedule (if no deadlock occurs),
   5.will necessarily result in a serial schedule (if no deadlock occurs), and
   6.may result in a deadlock.

   a) T1 = **L1**(C) w1(C) **L1**(A) r1(A) w1(A) **L1**(B) r1(B) w1(B) Commit **U1**(A) **U1** (C) **U1**(B)
   T2 = **L2**(B) r2(B) w2(B) **L2**(A) r2(A) w2(A) Commit **U2**(A) **U2**(B)
   b) T1 = **L1**(B) **L1**(C) w1(C) **L1**(A) r1(A) w1(A) r1(B) w1(B) Commit **U1** (A) **U1**(C) **U1**(B)
   T2 = **L2**(B) r2(B) w2(B) **L2**(A) r2(A) w2(A) Commit **U2**(A) **U2**(B)
   c) T1 = **L1**(C) **L1**(A) w1(C) r1(A) w1(A) **L1**(B) r1(B) w1(B) **U1**(A) **U1**(C) **U1**(B) Commit
   T2 = **L2**(B) r2(B) w2(B) **L2**(A) r2(A) w2(A) Commit **U2**(A) **U2**(B)

3. Consider again the two transactions above. Describe the schedule (in terms of read, validate, and write phases) where an optimistic concurrency control system can run these two transactions without aborting one of them.