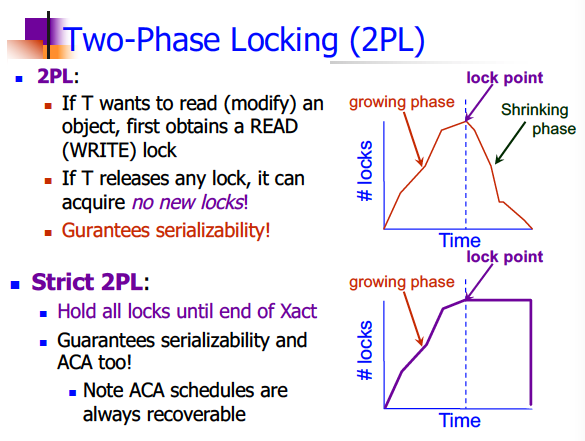
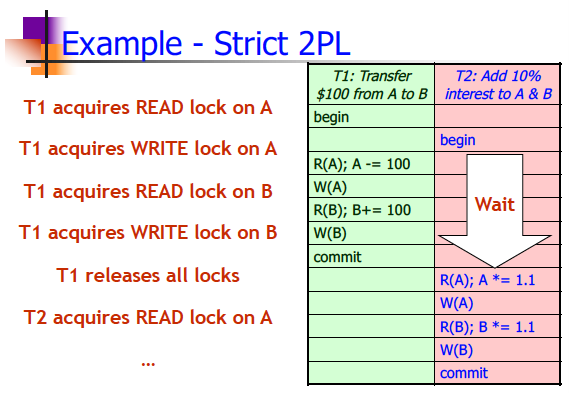
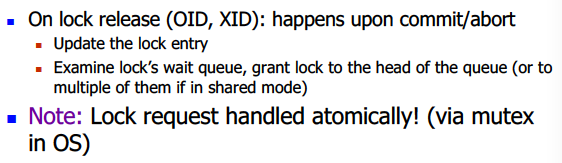
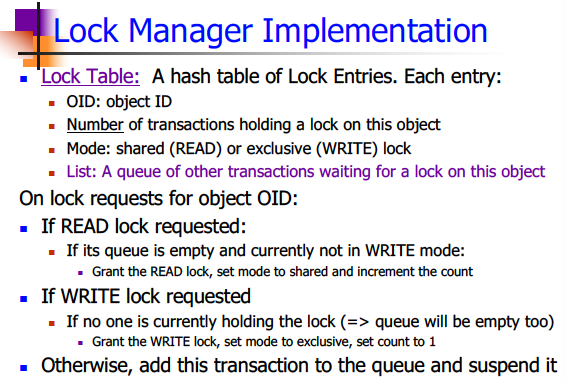
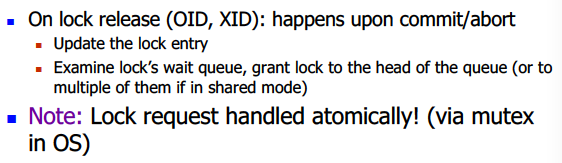
1. Lecture 10: Transaction Management
   1. ACID Properties
      1. Atomicity
         1. All or nothing property (transactions don’t execute partially)
      2. Consistency
         1. Assume that each submitted transaction, if executed, satisfies any integrity constraints
      3. Isolation
         1. Each transaction appears to execute without concurrent transactions (serially)
      4. Durability
         1. If a transaction commits, its effects persist
   2. Schedules
      1. Interleaving actions from a set or transactions
   3. Serial Schedule
      1. No interleaving actions among transactions
   4. Serializable Schedule
      1. Equivalent to a serial schedule. Can be transformed into a serial schedule by re-ordering the non-serial actions.
      2. Essential!!
      3. Different serial schedules can have different final states
   5. Conflicts
      1. Write-read (WR) conflict
         1. “Dirty Read”
         2. Read a page after it has been written to
      2. Read-write (RW) conflict
         1. “Unrepeatable read”
         2. T1 reads the value of object A, then T2 updates A (before T1 has committed)
      3. Write-write (WW) conflict
         1. Overwriting uncommitted data
         2. T2 overwrites a page that T1 wrote to before T1 was committed
   6. Aborts
      1. Makes the transaction “disappear”
   7. Cascading Abort
   8. Recoverable Schedule
      1. Transactions commit only after all transactions whose changes they have read commit.
      2. Avoid cascading aborts (ACA)
         1. Transactions read only the changes of committed transactions
   9. Locking
      1. 
      2. 
   10. Precedence (Serializability) Graph
       1. A node for each committed transaction in L
       2. An arc from T­i to Tj if some action in Ti precedes and conflicts with some action in Tj
   11. Conflict Equivalent
       1. If they involve the same transactions
       2. And order each pair of conflicting transactions in the same way
   12. Conflict Serializable
       1. If it is conflict equivalent to a serial schedule
       2. All CS schedules are also serializable.
       3. **A schedule is conflict serializable if and only if its precedence graph is acyclic**
   13. 2PL
       1. Guarantees an acyclic precedence graph
   14. Strict 2PL
       1. Guarantees ACA (read only committed values) by holding write locks until the end of a transaction
   15. Lock Manager Implementation
       1. 
   16. Deadlocks
       1. A cycle in the “Waits-for” Graph
       2. Prevention
          1. Assign Priorities to transactions
          2. 
   17. Non-Locking CC protocols
       1. Optimistic CC
       2. Multiversion CC
   18. 