Chapter 8 Questions

Questions may be asked in class, on quizzes, and on exams

* + Identify and describe the necessary and sufficient conditions for a deadlock to take place

**1. Mutual exclusion**. At least one resource must be held in a nonsharable mode; that is, only one thread at a time can use the resource. If another thread requests that resource, the requesting thread must be delayed until the resource has been released.

**2. Hold and wait**. A thread must be holding at least one resource and

waiting to acquire additional resources that are currently being held by

other threads.

**3. No preemption**. Resources cannot be preempted; that is, a resource can be released only voluntarily by the thread holding it, after that thread has completed its task.

**4. Circular wait**. A set *{T*0, *T*1, ..., *Tn}* of waiting threads must exist such that *T*0 is waiting for a resource held by *T*1, *T*1 is waiting for a resource held by *T*2, ..., *Tn*−1 is waiting for a resource held by *Tn*, and *Tn* is waiting for a resource held by *T*0.

* + Identify and describe the three commonly-used strategies for handling deadlock conditions

• We can ignore the problem altogether and pretend that deadlocks never occur in the system.

• We can use a protocol to prevent or avoid deadlocks, ensuring that the system will ***never*** enter a deadlocked state.

• We can allow the system to enter a deadlocked state, detect it, and recover.

* + Identify and explain two strategies for preventing deadlock

Mutual Exclusion：, at least one resource is not Shared.

Shareable resources do not require mutex access, so deadlocks are not involved

Hold and Wait： We must ensure that every time a thread requests a resource, it does not hold any other resources

* + Identify and explain the main strategy for avoiding deadlock
  + 1. safe state
  + 2. recourse
  + Given the condition(s) of a given system, identify a safe or potentially unsafe state in which the system may be in
  + Apply the Banker's algorithm to a given resource allocation and/or potential deadlock circumstances
  + Draw or apply the Resource Allocation graph to resource allocation and/or potential deadlock circumstances, for resources with single or individual instances
  + Identify and explain one strategy for managing recovery from a deadlocked condition
  + End of Chapter 8
    - Exercises: 8.1, 8.2, 8.3, 8.7, 8.9, 8.11, 8.12, 8.18, 8.20, 8.21, 8.22, 8.24, 8.27, 8.28
    - Programming Problem: none