資訊網路與多媒體研究所

98. 3. 6

資格考試科目：高等作業系統

Instructions: There are **six** questions which count 100 points in total. Each question may have several sub-questions. Please read the questions carefully before answering the questions.

1. There are two general approaches to deal with deadlocks. Please answer the following questions. (20 pts)
   A. Please define deadlock prevention and deadlock avoidance? (5pts)
   B. How do you ensure that starvation will not occur when dealing with deadlocks. (5pts)
   C. Alice designed a resource synchronization protocol. In her protocol, each program has a unique program ID and each resource also has a unique resource ID which is the greatest program ID among that for the programs may request the same resource. Bob's protocol is the following: A process can lock a resource if and only if its program ID is ***greater*** than the resource ID of other resources already locked by other processes. Alice claims that her protocol can avoid deadlock. Is Alice correct? If yes, please prove that there is no deadlock. If no, please use a counter example to prove that Alice's protocol cannot avoid deadlock. (10pts)

2. In distributed systems, it is very often that a coordinator is needed to synchronize the work among the processes in the systems. Election algorithms such as bully and ring-based algorithms are designed to determine which process is the coordinator process. Assume that there are *n* processes in the system and each process has a unique label. Please answer the following questions: (20pts)
   A. In the conventional ring-based election algorithm, a unidirectional ring is used. What is the number of messages needed in worst case and in average to elect a new coordinator when only the coordinator process fails? (10pts)
   B. In order to shorten the time to elect the new coordinator for ring-based election algorithm, Bob suggests to use a bidirectional ring. Can the election algorithm be made more efficient? If no, explain why. If yes, suggest such an algorithm and compare the number of messages needed for electing a coordinator in the two algorithms. (10pts)

3. Please answer the following questions for process management: (16pts)
   A. *pagedaemon* is a special process in Unix. What is its responsibility? Is it a long-term, mid-term, or short-term scheduler? You must explain why to receive any credits. (8pts)
   B. What is the major difference between vfork() and fork()? (10pts)

4. Please answer the following questions for process scheduling: (20pts)
   A. Consider N tasks being scheduled by round-robin scheduling, where tasks are all ready at time 0. Please compare the average completion time of the N tasks under round-robin scheduling and that under First-Come-First-Serve. How will round-robin scheduling become First-Come-First-Serve? (Hint: Please consider the context switching overhead.) (10pts)
   B. Consider non-preemptive priority scheduling of N tasks in a uniprocessor system, where there are M priority levels, and M divides N. Each task is given a fixed priority level when it is dispatched, and each priority level has N/M tasks. Tasks of the same priority are scheduled by First-Come-First-Serve. What is the maximum number of task executions for a task with the lowest priority? What is the maximum number of task executions for a task with the highest priority? (10pts)

5. Please answer the following questions for process synchronization: (14pts)
   A. Please define "binary semaphores". (4pts)
   B. Suppose that a code segment A can execute only if any one of the two code segments B and C finishes its execution. How to use binary semaphores to implement the requirement? You must write code with wait and signal operations to receive any credit. (10pts)

6. When the multi-core processors are used in a system and the physical memory are shared among all the cores, the operating system has to assure that the data are consistent all the time. One general approach is to organize all the memory in memory pages and lock the memory page when used. However, one drawback for this approach is *false sharing*. Please answer the following questions. (10 pts)
   A. Please define false sharing for shared memory. (4pts)
   B. Please describe the condition that false sharing may occur more frequently. and how can we avoid the false sharing? (6pts)