

資格考試科目：高等作業系統

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

1. (16 pts) In a multi-processor/core systems, shared memory can reduce the inter-process/core communication overhead. To assure data consistence for shared memory, consistency models are used to describe what level of data consistency are provided in the system.
 - A. (8 pts) Strict consistency model requires that all the read operations must read the value the mostly recent written value. Please describe the challenges for implementing this model on loosely coupled distributed systems?
 - B. (8 pts) Sequential consistency model requires that all the processes on the system see the same order of memory access. Please describe the rationale for not preserving the memory access order in consistent with actual memory access order.

2. (32 pts) Alice proposed a new scheduling algorithm and claims that the algorithm favors new processes and is fair to old processes. In Alice's scheduling algorithm, there are two queues: new process queue and old process queue. When a new process arrives, the process is inserted into the new process queue. The other ready processes are kept on the old process queue. When it is time to schedule, the scheduler always selects one new process if new process queue is not empty; otherwise, it selects one process from the head of old process queue. Each process will be given a 10-milliseconds time slot. (NOTE: You will receive only one point if you can only correctly answer the questions with YES or NO.)
 - A. (6 pts.) Is this algorithm starvation free? If yes, please explain your answer. If not, please give an example showing that a process may starve.
 - B. (6 pts.) The processes on old process queue may wait a long time to execute. One possible solution to shorten the waiting time is to move new processes to the old process queue after it stays on the queue for more than 100 milliseconds. Will this approach shorten the waiting time? Please argue your answer.
 - C. (6 pts) When the solution in (B) is implemented, is this algorithm starvation free? If yes, please explain your answer. If not, please give an example showing that a process may starve.
 - D. (6 pts) The mechanism of moving new processes to the old process queue can be implemented by setting timers for every process. However, it incurs heavy overhead to the system. Can this mechanism be implemented without timer? If Yes, please describe the solution. Otherwise, please argue it.
 - E. (8 pts.) Discuss whether algorithm (B) is fair to all processes? By fair, we mean that every process has a wait time approximately equal to the average wait time, assuming all processes have the same execution time. Please discuss for the following two cases.
 - i. (4 pts) Assume that the execution times of the processes are much less than 100 milliseconds.
 - ii. (4 pts) Assume that the execution times of the processes are much longer than 100 milliseconds.

3. Please answer the following questions for process management: (16 pts)
 - A. What is the major difference between vfork() and fork()? (4 pts)
 - B. In the shell environment, when an external command is provided, a child process is created to execute the external command. Should the child process be created by vfork() or fork()? Please argue your answer. (6 pts)

- C. Copy-On-Write (COW) technique is used to shorten the execution time for fork(). Can the same technique be adapted to implement vfork() to shorten its execution time? Please argue your answer. (6 pts)
4. (20 pts) Please answer the following questions for process synchronization:
- A. (6 pts) Please define **serializable schedule** and **serial schedule**.
 - B. (6 pts) Please define **two-phase locking** and **strict two-phase locking** for concurrency control.
 - C. (4 pts) Please explain the reason that the transaction schedules are serializable when two-phase locking is used for concurrency control.
 - D. (4 pts) Suppose that two processes, P_1 and P_2 , read and write a data object D . In other words, there are four locks, $R(P_1, D)$, $R(P_2, D)$, $W(P_1, D)$, and $W(P_2, D)$. Please provide an example schedule causing deadlock when two-phase locking is used.
5. (16 pts) When multi-core processors are used in a system and computation workload are shared among all the cores, the operating system usually tries to evenly dispatch the tasks among the cores. Please answer the following questions.
- A. (4 pts) Please define **load balance** and **load sharing**.
 - B. (6 pts) One approach for load balance is to maintain a global task queue for all the ready tasks and use one of the cores to dispatch the tasks. Please describe the advantage and disadvantage of this approach.
 - C. (6 pts) To conduct load sharing/load balance, the operating system should be aware of the load on each core. One may use the average task response time or length of the task queue to estimate the load. Please compare the accuracy and runtime overhead of these two approaches.