1. (20 pts) Different CPU scheduling algorithms are needed in different application environments. For example, batch systems emphasize the maximum of throughput while interactive systems emphasize response time. Consider four common scheduling algorithms, including First-come, first-served (FCFS) scheduling, shortest-job-first (SJF) scheduling, round-robin (RR) scheduling, and priority scheduling algorithms. Please answer the following questions.
   a. (4 pts) Which scheduling algorithms are appropriate for batch systems?
   b. (4 pts) Which scheduling algorithms are appropriate for interactive systems?
   c. (4 pts) Which scheduling algorithms may result in starvation?
   d. (8 pts) Consider the following four processes.

Process	Arrival Time	CPU Time	Priority
P1	0	20	2
P2	4	10	1
P3	6	15	1
P4	10	5	2

   What is the average waiting time for FCFS, preemptive SJF, RR, and nonpreemptive priority algorithms, respectively? Assume that a low number implies a higher priority, the time quantum for RR is 2, and the time for context switch is ignored.

2. (22 pts) Given a computer system with *m* resources shared by *n* processes, please answer the following questions.
   a. (6 pts) Assume the *m* resources are of the same type, *m*=10 and *n*=3. Give an example of unsafe states by completing the following table. Justify your answer.

	Maximum needs	Current needs	(Free: _____)
P1	_____	_____	
P2	_____	_____	
P3	_____	_____	

   b. (8 pts) If the *m* resources are all different and ordered numerically (from 1 to *m*), please explain why that each of the *n* processes requests the resources in an increasing order of enumeration can prevent deadlock. What necessary condition cannot hold here for a deadlock to occur?
   c. (8 pts) Assume the *m* resources are of the same type such as printers and each process needs at most *k* resources. *m*=12 and *n*=5. What is maximum value of *k* such that the system is deadlock-free? Explain your answer.

3. (20 pts) Spin locking is often used to implement semaphores but has the defect of requiring busy waiting, which wastes CPU cycles.
    a. (6 pts) Please give an advantage of a spin lock and explain why it is useful in multiprocessor systems?
    b. (6 pts) Discuss if a spin lock works when CPU scheduling is preemptive and when CPU scheduling is nonpreemptive.
    c. (8 pts) Describe how to modify the two atomic operations, wait() and signal(), to overcome the need for busy waiting.

4. (18 pts) Consider five disk-scheduling algorithms, including FCFS scheduling, SSTF scheduling, SCAN scheduling, C-SCAN scheduling, and LOOK scheduling algorithms. Please choose a best scheduling algorithm for the following questions.
    a. (6 pts) Which algorithm is truly fair? Explain your answer.
    b. (6 pts) Which algorithm is the best for heavily loaded systems? Explain your answer.
    c. (6 pts) Which algorithm is the best for those systems where I/O requests are not uniformly distributed, e.g., system files may be accessed more frequently? What is the drawback of this algorithm? Explain your answer.

5. (20 pts) Consider the timestamp-ordering approach with resource preemption for deadlock prevention in a distributed system, where a unique priority number is assigned to each process and then used to decide whether a process should wait for another one or just roll back.
    a. (5 pts) Explain why the timestamp-ordering approach may result in starvation.
    b. (5 pts) Globally unique timestamps can be generated using either a centralized or distributed approach. Please compare the two approaches.
    c. (5 pts) Please define the *wait-die* and *would-wait* schemes, the schemes can avoid starvation.
    d. (5 pts) Discuss how the two schemes perform better under different circumstances.