Problem 1: (12 points; 10 minutes) Briefly explain any 3 of the following concepts

1. Programmed I/O
2. Conditional Critical Regions
3. Memory hierarchy
4. Atomic instruction

Problem 2: (18 pts; 12 minutes)

(a) (4 points) Why are system calls needed?
(b) (4 points) Which is more expensive a system call or a procedure call; why?
(c) (5 points) Consider multiple threads running in the same space (in the same process). What happens if the process is swapped while thread 1 is running?
(d) (5 points) For a process with multiple threads, what happens when one thread waits on a semaphore whose value is 0?

Problem 3: (12 points; 8 minutes) Consider process scheduling on a system with multiple processors, each of which can run a process.

(a) (3 points) How would the process state diagram change?
(b) (9 points) When it is time to schedule a new process, a processor runs the process that is at the top of the ready queue. Explain what problems may occur if two processors need to switch to a new process at the same time. How would you fix such problems?

Problem 4: (15 points; 12 minutes) Explain three of the following potentially wrong statements.

1. Peterson’s algorithm provides a fair blocking lock
2. An acceptable solution to the critical region problem must ensure mutual exclusion, concurrency and correctness
3. Semaphores can be implemented using locks
4. A thread switch is less expensive than a process switch only if the two threads are in the same process

Problem 5: (26 pts; 20 minutes) A mini-bus shuttles between three stops around the university circle going in the same direction. Passengers wait at their stop until the bus arrives. The bus has seats for 10 passengers. When the bus reaches a stop, the passengers going down leave the bus, and the new passengers climb on the bus if there is room (assume they leave from different doors). The bus driver waits for all the passengers to climb, before moving to the next station. You may use semaphores, locks and condition locks.

(a) (13 points) Write pseudocode to simulate this bus operation.
(b) (13 points) Assume that the bus has room for 5 standing passengers as well. A passenger sits if a seat is available, otherwise she stands. If a seat becomes available later, she sits down (however, if her station comes up first, she just leaves the bus). Write pseudocode to simulate this situation.

Problem 6: (17 pts; 12 minutes) Consider the following versions of the producer consumer problem. They differ in the mechanism used for implementing the critical region as well as what is included in the critical region: (1) A version that uses Peterson’s algorithm to restrict access around the shared buffer; (2) A version that disables interrupts around the access of the shared buffer; (3) A version that uses a binary semaphore around the access of the shared buffer; and (4) a version that uses semaphores around the full consumer and producer code.

(a) (8 points) Show an implementation for version (3) with an unbounded buffer.
(b) (9 points) Explain which of these versions will be most efficient.