Operating Systems (COMP2006)

CURTIN UNIVERSITY
Discipline of Computing
School of Electrical Engineering, Computing and Mathematical Sciences

Worksheet 3

1. What is a critical section? What is the critical-section problem?

2. What three requirements must a solution to the critical-section problem satisfy? Explain the meaning of each requirement

3. What is an *atomic* instruction? Show that if the *wait* operation is not executed atomically, then mutual exclusion may be violated

4. The first known correct software solution to the critical-section problem for two processes was developed by Dekker. The two processes, P_0 and P_1, share the following variables:

```
var flag: array [0..1] of boolean;  /* initially false */
        turn: 0..1;
```

The structure of process P_i (i = 0 or 1), with P_j (j = 0 or 1) being the other process is shown as follows

```
repeat
   flag[i] := true;
   while flag[j] do
   begin
      if turn = j then
      begin
         flag[i] := false;
         while turn = j do no-op;
         flag[i] := true;
      end;
   end;

   critical section
   turn := j;
   flag[i] := false;
   remainder section
until false;
```

Prove that the algorithm satisfies all three requirements for the critical section problem.

5. What is spinlock? Discuss the advantage and disadvantage of using spinlock. Why do you think Solaris, Linux, and Windows 2000 use spinlocks as a synchronization mechanism only on multiprocessor systems and not on single- processor systems?

6. Suppose we want to execute statements S_1, S_2, and S_3 in sequence, but that S_2 has to be executed exclusively for one process at a time. Write the code needed using semaphores

7. Explain when and how to use each of the following functions:

   - Pthread_Mutex_Lock()
   - Pthread_Mutex_Unlock()
   - Pthread_Cond_Wait()
   - Pthread_Cond_Signal()

8. *Sleeping Barber* problem. Consider a barbershop with one barber, one barber chair, and *n* chairs for waiting customers, if any, to sit in. If there are no customers present, the barber sits down in the barber chair and falls asleep. When a customer arrives, he has to wake up the sleeping barber. If additional customers arrive while the barber is cutting a customer's hair, they either sit down (if there are empty chairs) or leave the shop (if all chairs are full). The problem is to program the barber and the customers without getting into race conditions.