COS214 Tutorial 9

Roberto Togneri, 2000

1. (a) Imagine there's a twin track railway line with a single track section going through a tunnel. How would you design a system that would ensure that only one train at a time enters the tunnel in any one direction? What are the attributes of your solution? For example, does this allow trains from one direction through even it there are no trains from the other direction?

   (b) In a restaurant there is a seating area for patrons with a narrow corridor (with a swing door at each end) for the waiters to gain access to the kitchens. It is so narrow that it will only allow for a waiter to go through it in one direction at a time. The manager installed a solution to the problem of collisions - a switch at each end that put a light on at the other end. If the light was on then you didn't enter the corridor. If the light was off, then you threw the switch, travelled through and then threw the switch at the other end to turn the light off. This worked fine, but yet there were still complaints with occasional accidents. The waiters complained that the system was unfair. Can you explain why there were still accidents and why the waiters complained?

2. (a) What is meant by the term concurrent processing?
   (b) Give a definition of mutual exclusion. Give examples of where mutual exclusion is required for proper operation.
   (c) What is a race condition? What is a critical region?
   (d) What is meant by the term "deadlock"?

3. For the following program fragments *assume that ProcessNumber is a shared integer variable initialised to 1; P1Inside,* P2Inside are shared Boolean variables initialised to false; and *P1WantsToEnter* and *P2WantsToEnter* are shared Boolean variables initialised to false. Further, assume each process is in some repeating loop and so continually, but at possibly differing rates, re-executes the code shown. For each of the three cases explain why they are unsatisfactory as solutions to the mutual exclusion problem.

Process One	Process Two
(a) while(ProcessNumber==2); CriticalSection(); ProcessNumber=2; …	(a) while(ProcessNumber==1); CriticalSection(); ProcessNumber=1; …
(b) while(P2Inside); P1Inside=TRUE; CriticalSection(); P1Inside=FALSE; …	(b) while(P1Inside); P2Inside=TRUE; CriticalSection(); P2Inside=FALSE; …
(c) P1WantsToEnter=TRUE; while(P2WantsToEnter); CriticalSection(); P1WantsToEnter=FALSE; …	(c) P2WantsToEnter=TRUE; while(P1WantsToEnter); CriticalSection(); P2WantsToEnter=FALSE; …

4. You are designing a mutual exclusion primitive for a computer that does not have a test and set instruction. Instead it has an instruction that interchanges the contents of a register with a memory location in a single atomic action. Use this instruction to create primitives to enter and leave critical regions.

5. From the earlier tutorial that gave the problem of the waiters in a restaurant colliding in a one way tunnel between the eating area and the kitchen, can you now think of how to solve this problem using a simple semaphore solution? Describe any limitations of your solution.

6. A road bridge has only a single lane. To the North and South are feeder roads that are two laned. More than one car can be on the bridge if travelling in the same direction. Using semaphores, write the pseudo-code routines *enter_south*( ), *leave_south*( ), *enter_north*( ), *leave_north*( ) that arrange for cars to safely cross the bridge [**Hint:** Readers & Writers semaphore solution]. What are the attributes and problems of your proposed solution?