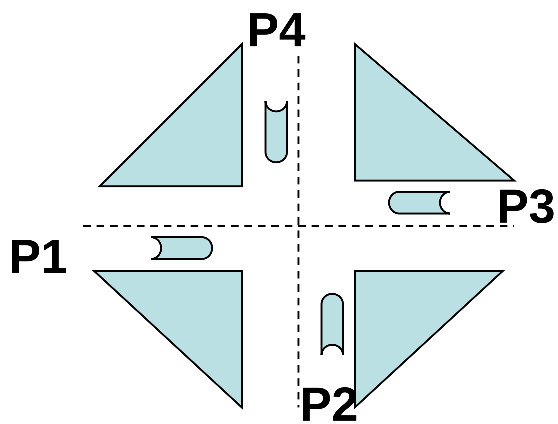
Homework Assignment 3

1. Use semaphores to solve the crossroads' problem . In this problem, the street is a two-way street. So P1 and P3 can pass through the intersection simultaneously. But the cars in the horizontal and the cars in the vertical direction cannot pass through the intersection at the same time.



P1	P2	P3	P4
{	{	{	{
pass the crossroad;	pass the crossroad ;	pass the crossroad ;	pass the crossroad;
}	}	}	}

2. A group of students are studying for a OS exam. The students can study only while eating pizza. Each student executes the following loop: *while (true) { pick up a piece of pizza; study while eating the pizza}*. If a student finds that the pizza is gone, the student goes to sleep until another pizza arrives. The first student to discover that the group is out of pizza phones PizzaShop to order another pizza before going to sleep. Each pizza has *S* slices.

Write code to synchronize the student threads and the pizza delivery thread. Your solution should avoid deadlock and phone PizzaShop (i.e., wake up the delivery thread) exactly once each time a pizza is exhausted. No piece of pizza may be consumed by more than one student.

3. The Sleeping-Barber Problem. A barbershop consists of awaiting room with n chairs and a barber room with one barber chair. If there are no customers to be served, the barber goes to sleep. If a customer enters the barbershop and all chairs are occupied, then the customer leaves the shop. If the barber is busy but chairs are available, then the customer sits in one of the free chairs. If the barber is asleep, the customer wakes up the barber. Write pseudo code to coordinate the barber and the customers.

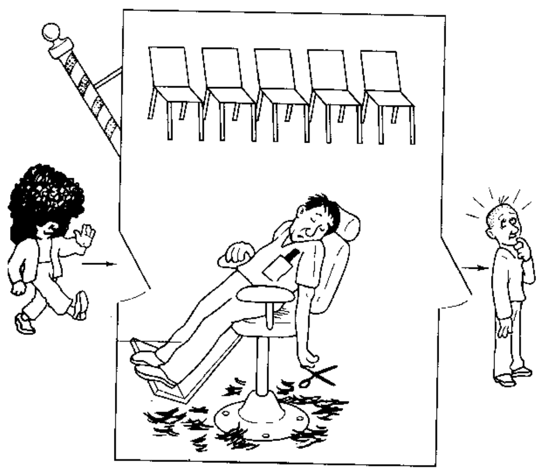


Figure 2-20. The sleeping barber.

4. Write a no deadlock version of pseudo code for dining-philosophers problem.

The structure of philosopher process i :

```
do {
     wait (chopstick[i] );
      wait (chopStick[ (i + 1) % 5] );

                    //  eat

      signal (chopstick[i] );
      signal (chopstick[ (i + 1) % 5] );

                    //  think


} while (TRUE);
```

5. Consider the following resource allocation graph:

P = {P1,P2,P3,P4}

R = {R1,R2,R3}

E = {R1 --> P1, P1 --> R2, R2 --> P2, P2 --> R3, R3 --> P3, P3 --> R1, P4 --> R1}

- resource type R1 has one instances
- resource type R2 has one instance
- resource type R3 has one instance

Draw a resource allocation graph and explain the possibility for a deadlock.

6 .Consider the following T0 snapshot of a system:

	Max	Allocation	Available
	ABC	ABC	ABC
P0	5 5 9	2 1 2	2 3 3
P1	5 3 6	4 0 2	
P2	4 0 11	4 0 5	
P3	4 2 5	2 0 4	
P4	4 2 4	3 1 4	

Answer the following questions using the banker's algorithm:

a) What is the content of the matrix Need?

b) Is the system in a safe state?

c) If a request from process P2 arrives for (0,3,4), can the request be granted immediately?

d) If a request from process P4 arrives for (2,0,1), can the request be granted immediately?

Briefly Explain.