

### **Sleeping Barber Problem:**

In this project, it is aimed to find a solution to the sleeping barber problem, which is a synchronization problem. In solution three semaphores and one mutex are used.

- Semaphore cus counts waiting customers
- Semaphore bar is the number of available barbers (0 or 1)
- Semaphore seats is for modifying seat number.
- Mutex cutting represent the one seat and the barber is busy

There is one thread for barber, and n thread for customers. Each calls its own function.

At that point I could not handle with the customer and barber synchronization, therefore the code is not fully correct. It cannot calculate the separate waiting times, but calculate the total waiting time. If each customer goes to barber <haircut\_repetition > times, the code terminates. Also in inputs extra input as added for number of seats.

If barber is working, it locks the cutting mutex and its semaphore.

If a customer gets haircut, it locks the barber, and seat semaphore. When operation is done according to the calculated times, the locked semaphores are released.