

# The Sleeping-Barber Problem

- Conditions:
  - **1** waiting room with  **$N$**  chairs.
  - **1** barber room with **1** barber chair
- Barber and Customers
  - If there is no customers, the barber goes to sleep.
  - If a customer enters the shop
    - If all chairs are occupied, the customer leaves.
    - If the barber is busy and chairs are available, the customer sits in one of the free chairs.
    - If the barber is asleep, the customer wakes up the barber.

# Variables

- Semaphore Customers = 0;
- Semaphore Barber = 0;
- Mutex accessSeats = 1;
- int NumberOfFreeSeats = N;  
//total number of seats

# Barber and Customers

## Barber

```
while(1) {  
  P(Customers) //wait for C and sleep  
  P(accessSeats) //mutex protect the number of  
    available seats  
  NumberOfFreeSeats++ //one chair gets free  
  V(Barber) //Bring in a C for haircut  
  V(accessSeats) //release the mutex on the  
    chairs  
  ..... //here the B is cutting hair  
  
} //while(1)
```

## Customers

```
while(1) {  
  P(accessSeats) //mutex protect the number of  
    available seats  
  if ( NumberOfFreeSeats > 0 ) { //if any free  
    seats  
    NumberOfFreeSeats-- //sitting down on a  
    chair  
    V(Customers) //notify the B  
    V(accessSeats) //release the lock  
    P(Barber) //wait if the B is busy  
    .... //here the C is having his hair cut  
  } else { //there are no free seats  
    V(accessSeats) //release the lock on the seats  
    //C leaves without a haircut  
  }  
} //while(1)
```