CS 511 – Quiz 3: Semaphores

19 September 2018

Names:
Pledge:

A barbershop[1] consists of a waiting room with n chairs, and the barber room containing the 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 (you may assume you have a function `doneWithThisBarber()` that never returns). 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. Once the barber finishes helping a customer, the customer pays and then leaves. Write a program to coordinate the barber and the customers.

Hint: You need only use one counter, for customers, in order to determine whether the barbershop is full.

```
// declarations
int customers=0;
      semaphore customer, barbersAttention, permissionToLeave, paymentCompleted = new semaphore(0);
      semaphore mutex = new semaphore(1);


thread Customer: {
       mutex.acquire();
       if customers >= N
         mutex.release();
         doneWithThisBarber();
       customers++;
       mutex.release();
       customer.release();
       // waits for barbersAttention to get haircut
       barbersAttention.acquire();
       // gets haricut
       // waits for barbersAttention to pay them
       payment.acquire();
       // pays barber
       // waits for barbersAttention to leave
       permissionToLeave.acquire();
       mutex.acquire();
       customers--;
       mutex.release();
       //leaves
}

thread Barber: {
    while(true) {
      // rendezvous with a customer
      customer.acquire();
      barbersAttention.release();




       // cut hair
       // rendezvous for signalling end of cut
       payment.release();
       // receives pay from customer
       permissionToLeave.release();
       // allows customer to leave



  }
}
```

[1] Originally proposed by Dijkstra