

OS (CSL373/CS333N)

Minor I

February 25, 2006

Maximum marks : 60

Name: _____ Entry: _____ Grp: _____

The following is a version of the *sleeping barber problem*. Several OS books have discussions on variants of this problem.

Assume that the *SAC* barbershop has three chairs, three barbers, and a waiting area that has four *Filmfare* magazines and an unlimited supply of *S+G OS* books for other customers. *SAC* regulations limit the total number of customers in the shop to 20.

A customer should not enter the shop if it is filled to capacity with other customers. Once inside, if no barber is free then the customer grabs a *Filmfare* if available or a *S+G OS* book otherwise (essentially sleeps). When a barber is free, the customer who has been reading *Filmfare* the longest is served and, if there are customers studying *S+G OS* (sleeping), the one who has been studying for the longest time now gets to read *Filmfare*. When a customer's hair-cut is finished, any barber can accept payment, but because there is only one cash register, payment is accepted for one customer at a time. The barbers divide their time between cutting hair, accepting payment, and sleeping in their chair waiting for a customer (they are not interested in either *Filmfare* or *OS*). No customer should sit on the lap of another (improper aggressive behaviour) and the barbers should be fair - if one barber is very fast or one of the customers is quite bald, then a customer who has only had a partial hair cut should not be evicted from his seat and made to pay, neither should the bald guy be restrained in his seat even though his hair cut is complete. Implement processes `customer`, `barber` and `cashcounter` and synchronize using the following semaphores.

```
semaphore max_capacity = 20, filmfare = 4;
semaphore barberchair = 3, barberfree = 3;
/* intended usage of the above are obvious */
semaphore cust_ready=0,leavechair=0,payment=0,receipt=0;
/*barber waits/sleeps on cust_ready till a customer is in the chair,
  barber waits on leavechair till customer gets up from the chair,
  cashcounter waits on payment for a customer to pay,
  customer waits on receipt for a receipt from the cashcounter */
semaphore finished[50];
/*initialized to 0. customer n waits on finished[n]
  until his haircut is complete.
  assume a maxm of 50 customers on a fine sunday morning */

/* anything else you may wish to define */
```

Assume that the following procedures are available: `entershop`, `readfilmfare`, `sitinbarberchair`, `enqueue(customerno)`, `dequeue(customerno)`, `leavebarberchair`, `pay`, `exitshop`, `cuthair`, `acceptpayment` (and anything else you may wish to define).

Write the code corresponding to the three processes. You are free not to use the semaphores and procedures defined above and define some of your own. Please donot despair and pay attention to details. Please donot write junk just to fill up the pages.

Clearly indicate (within comments) what each semaphore is for. Also, clearly indicate the shortcomings of your solution if any - every unidentified shortcoming will definitely be penalized.