

Computing Systems Lab

IT69101
(Autumn 2010)

Assignment #7

20/10/2010

1. a). Implement semaphore operation $P(S)$ and $V(S)$.

[Hint: Use `<sys/ipc.h>`, `<sys/sem.h>` and `<sys/shm.h>` C-library functions in your implementation.]

- (b) Apply your semaphore solution to following critical section problem [producer-consumer problem]

```
PARBEGIN
  PRODUCER
    BEGIN
      REPEAT
        Produce an item in nextP
        While Counter = n do skip
        Buffer [in] = nextP
        in = (in + 1) mod n
        Counter = Counter +1
      UNTIL FALSE
    END
  CONSUMER
    BEGIN
      REPEAT
        While Counter = 0 do skip
        nextC = Buffer [out]
        out = (out +1) mod n
        Counter = Counter -1
        Consume the item in nextC
      UNTIL FALSE
    END
PAREND
```

- Note:** 1) The variables and statements in the pseudo code bear usual meaning.
2) You should identify the critical section in the code.
3) Use the join-fork constructs to run the producer-consumer concurrently.
4) Assume the proper initialization of the variable in the main routine. Assume the buffer size with a moderate value, $n = 3, 4, 10$ etc.
5) Run the program to ensure that your implementation satisfies i) concurrency, ii) progress condition and iii) mutual exclusion.

- c) Modify your implementation to avoid “*busy waiting*” problem in the above solution.
- d) Repeat the problem 1(b) to solve *m-Producer* and *n-Consumer* problem ($m, n \geq 1$).

2. a) Maintain n objects ($n \geq 10$) for n customers with the following attributes

Customer:

```
customerNo:int  
balance:float  
withdrawAmount(amount:float)  
depositAmount(amount:float)  
displayAmount()
```

- b) Using multi-threading concept, run all instances concurrently.
- c) Test your implementation i) without synchronization and ii) with synchronization and report results in each case.

[**Hint**: You should implement Problem 2 in Java]

Last date of submission: 03/11/2010