Centre for Advance Studies

Dr. APJ Abdul Kalam Technical University

Secure Software Design and Operating System Security
(Code: MCySC-202)

Date Thursday 1st February, 2018

Time: 10:30 AM- 12:30AM **Lab 3** **Lab type: Evaluative (10 maximum marks)**

Race condition security

A company produces product a X product and have available **available** number of items to sell. Consider a function **producerConsumer(int quantity)**

```
1
2  void producerConsumer( int quantity)
3  {
4     int i,temp;
5     for(i=0;i<50000;i++)
6     {
7     temp= available; // available is a static global variable
8     temp=temp+quantity;
9     available=temp;
10    }
11 }
```

- if quantity is in positive then the call of producerConsumer(int quantity) is for Producer
- else if quantity is in negative then the call of producerConsumer(int quantity) is for Consumer

The company use to Produce 50000 item first and them sell them all to consumers in a day from long back till now. Consider the following code for the same.

```
1  static int available; // initially availability  is zero
2  void main( )
3  {
4     producerConsumer( 1);
5     producerConsumer( -1);
6     printf("\n The Availability of items are", available);
7  }
```

a Now the CEO of the company want the production and sales should be done simultaneously i.e.producer and consumer should work together concurrently. So write a code to execute producerConsumer simultaneously for producer and consumer as well. Then state the result of availability after the day.

b Did you find any problem in previous working, if yes then solve the problem with your approach and trace the time take by the approach and the previously used approach (without using concurrency).
Use **time** command to find execution time.