Vishwakarma Institute of Information Technology, Pune

Department of Computer Engineering

Continuous Evaluation – II

Submission Deadline – 14th April 2020 till 5 pm

Class: - T.Y. BTech **Subject: - Operating Systems**

Unit IV and V

1) Several processes wish to read and write data shared between them. Some processes only want to read, while others want to update the shared data. Multiple readers may concurrently access the data. However, a writer must not access the data concurrently with anyone else, either a reader or a writer. Write down pseudocode to implement the functions readLock, readUnlock, writeLock, and writeUnlock that are invoked by the readers and writers to realize read/write locks. You must use semaphores and mutexes only in your solution. **(02 Marks)**

2) Consider the following classical synchronization problem called the barbershop problem. A barbershop consists of a room with N chairs. If a customer enters the barbershop and all chairs are occupied, then the customer leaves the shop. If the barber is busy, but chairs are available, then the customer sits in one of the free chairs and awaits his turn. The barber moves onto the next waiting seated customer after he finishes one hair cut. If there are no customers to be served, the barber goes to sleep. If the barber is asleep when a customer arrives, the customer wakes up the barber to give him a hair cut. A waiting customer vacates his chair after his hair cut completes. Your goal is to write the pseudocode for the customer and barber threads below with suitable synchronization. You must use only semaphores to solve this problem. Use the standard notation of invoking up/down functions on a semaphore variable.

The following variables (3 semaphores and a count) are provided to you for your solution. You must use these variables and declare any additional variables if required.

semaphore mutex = 1, customers = 0, barber = 0; int waiting_count = 0;

Some functions to invoke in your customer and barber threads are:

- A customer who finds the waiting room full should call the function *leave ()* to exit the shop permanently. This function does not return.
- A customer should invoke the function *getHairCut ()* in order to get his hair cut. This function returns when the hair cut completes.
- The barber thread should call *cutHair()* to give a haircut. When the barber invokes this function, there should be exactly one customer invoking *getHairCut()* concurrently**.** **(03 Marks)**

OR

2) Consider the famous dining philosophers' problem. N philosophers are sitting around a table with N forks between them. Each philosopher must pick up both forks on her left and right before she can start eating. If each philosopher first picks the fork on her left (or right), then all will deadlock while waiting for the other fork. The goal is to come up with an algorithm that lets all philosophers eat, without deadlock or starvation. Write a solution to this problem using condition **variables or semaphores.** **(03 Marks)**

3) Answer yes/no and provide a brief explanation**.** **(01 Mark)**

(a) Is it necessary for threads in a process to have separate stacks?

(b) Is it necessary for threads in a process to have separate copies of the program executable?

4) A soft link can create a link between files across different file systems, whereas a hard link can only create links between a directory and a file within the same file system. [T/F] **(01 Mark)**

5) Consider the following: (03 Marks)

- **Work requests Queue for sectors: 23, 89, 132, 42, 187**
- **there are 200 cylinders numbered from 0 - 199**
- **the disk head stars at number 100**

Show the disk head movements and calculate the total time estimated for total arm motion using following disk scheduling policies

1) SSTF 2) SCAN 3) C- SCAN 4) SPTF