



**Laboratory Manual**  
*for*  
**Operating Systems Lab**

**(CL-2006)**

Course Instructor Ms Rubab Anam
Lab Instructor Muhammad Faheem
Section BCS-4F
Semester Spring 2025

Department of Computer Science  
FAST-NU, Lahore, Pakistan

**Objectives**

In this lab, students will practice synchronization problems. Please note that there should be no

deadlocks and there must be bounded waiting. You can do this lab either using processes or by threads.

Please read through `sem_overview` for overview of semaphore system calls in Linux. You need to use `sem_wait` and `sem_post` for waiting and signalling on a semaphore respectively. For named semaphores `sem_open` and `sem_close`, `sem_unlink` calls are used for creating and closing semaphores. For unnamed semaphores `sem_init` and `sem_destroy` perform similar operations.

## Lab Questions

### Question 1 (5)

Write a program to implement the sleeping barber problem. In this computer systems analogy, think about when a barbershop only has one barber: one barber chair and any number of chairs where customers wait. When there are no customers, the barber goes to sleep in their barber chair and must be woken when a new customer comes in. While the barber is cutting hair, new customers can take the empty seats to wait, or leave if there is no vacancy. Use appropriate semaphores to signal a barber thread once a new customer comes. Assume you have 5 waiting chairs and an additional chair for cutting hair. So at most there can be 5 waiting customers and an active customer in the barber shop.

### Question 2(5)

Write a program to solve the reader writer problem where there is a single writer W and two readers R1 and R2. Make sure that:

- the two reader threads can simultaneously access a shared section called data (eg char array) if W is currently not accessing.
- The writer must have a bounded wait, meaning if W comes before R2 in the waiting queue then it must be given access before R2.
- W thread can only access if both readers are not present.