School of Computer Science Engineering and Technology

Course- BTech	Type- Core
Course Code- CSET209	Course Name- Operating Systems (OS)
Year- 2023	Semester- Even
Date- 20/03/2023	Batch- 2021-2025

Lab Assignment 8

Exp. No.	Name	CO1	CO2	CO3
08	Synchronization	✓	✓	--

Objective: - The prime goal of this lab session is to implement synchronization between multiple concurrent processes.

Introduction: Suppose If three threads want to write a single file at the same time, it would be a problem because a hard drive cannot go as fast as a CPU does. Doing multiple operations on one target at the same time can lead to an inconsistent state. So, for these types of situations, we should lock the other threads in order to not overload the hard drive with tasks that can corrupt the file.

Task 1: Threads in a particular program share the same data segment and global and static variables are stored in the data segment. This implies that both global and static types of variables are shared by all the threads created.

Create a program to create five threads for a single program that shares the same data segment which is being used to store the global and static variables in the program. Initialize the global and static as integer types with values 0 and 1, respectively. At last, print the thread id with the values of static and global variables after incrementing it as mentioned in the sample output.

Note: Thread id may be different for everyone, but the values of global and static variables will be same.

Task 2: Semaphore is a data handling technique which is very useful in process synchronization and multithreading. In C programming we must include import<semaphore.h> as a header file to access the functionality of semaphore functions. Write a program in C to acquire the lock on a shared data item using semaphores to ensure the multithreading.
Hint: You can implement the same by using sem_wait() and sem_post() functions defined in semaphore.h header file for wait and signal operations, respectively as discussed in the lectures.