MES Wadia College of Engineering Pune-01

Department of Computer Engineering

Name of Student:	Class:
Semester/Year:	Roll No:
Date of Performance:	Date of Submission:
Examined By:	Experiment No: Part B-04

PART: C ASSIGNMENT NO: B-04

AIM: Write a program to solve Classical Problems of Synchronization using Mutex and Semaphore.

OBJECTIVES:

- To study the basics of classical problems of synchronization using Mutex and semaphore.

PRE-REQUISITES:

1. Basic of synchronization concepts.

APPARATUS:

THEORY:

Readers-writers problem relates to an object such as a file that is shared between multiple processes. Some of these processes are readers i.e. they only want to read the data from the object and some of the processes are writers i.e. they want to write into the object.

readers-writers problem is used to manage synchronization so that there are no problems with the object data. For example - If two readers access the object at the same time there is no problem. However if two writers or a reader and writer access the object at the same time, there may be problems.

To solve this situation, a writer should get exclusive access to an object i.e. when a writer is accessing the object, no reader or writer may access it. However, multiple readers can access the object at the same time. This can be implemented using semaphores.

The Readers-Writers Problem

A data item such as a file is shared among several processes.

Each process is classified as either a *reader* or *writer*.

Multiple readers may access the file simultaneously.

A writer must have exclusive access (i.e., cannot share with either a reader or another writer).

A solution gives priority to either readers or writers.

- *readers' priority*: no reader is kept waiting unless a writer has already obtained permission to access the database
- *writers' priority*: if a writer is waiting to access the database, no new readers can start reading

A solution to either version may cause starvation

- in the readers' priority version, writers may starve
- in the writers' priority version, readers may starve

CONCLUSION:

QUESTIONS:

1. What is mutex and semaphore?

2. Explain types of semaphore?

3. Explain the deadlock concepts?