**Department of Computer Science and Engineering**

|  |  |
| --- | --- |
| **Course Code: CSE 321** | **Credits: 1.5** |
| **Course Name: Operating Systems** | **Semester: Fall 18** |

**Lab 10  
Deadlock (Banker’s Algorithm)**

1. **Overview:**

A **deadlock** is a situation in which two or more competing actions are each waiting for the other to finish, and thus neither ever does. To prevent this situation Operating system introduces a well-known algorithm named Banker’s algorithm. This algorithm provides a nice solution for computer resource management by avoiding deadlock

Whenever a process requests resources,

* + the algorithm check if it accepting the request could put the system in an **unsafe state** (one where deadlock could occur).
  + If in unsafe state it deny request.

1. **Lesson Fit:**

Programming knowledge and deadlock concept is required for this lab.

1. **Learning Outcome:**

After this lab, students will know how deadlock would solve using banker’s algorithm even their own algorithm.

1. **Anticipated Challenges and Possible Solutions**
   1. Switching Process to process may occur problem for lacking matrix array concept..

**Solutions:** Students should have the knowledge about two dimensional array.

1. **Acceptance and Evaluation**

Students will show their progress as they complete each task. They will be marked according to their lab performance.

**Activity Detail**

* 1. **Hour: 1  
     Discussion:**

1. Discussion on Deadlock.
2. Why deadlock occur. How we can avoid deadlock for the list of coming process to CPU.
   1. **Hour: 2**
3. Discuss on Max, Allocation and Need Matrix of the recourse for the given processes.
   1. **Hour: 3**
4. Discuss on programming logic of Bankers algorithm.
5. Implement Bankers algorithm
6. Find Safe sequence for the given process.
7. Importance and Complexity of Bankers algorithm.
8. **Home tasks**

Create five processes with different resource requirement and find Safe sequence using Deadlock algorithm.

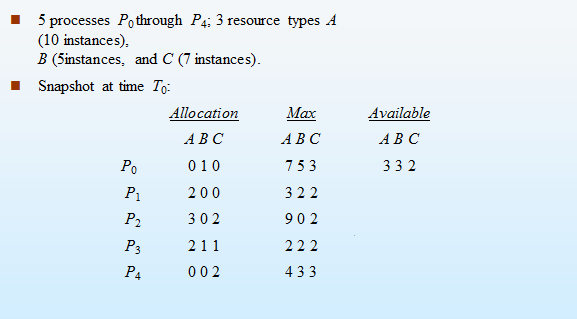
**Lab Activity List**

**Task 1**

Implement Banker’s Algorithm.

* Input:
  + Number of process
  + Number of resource type
  + Number of resource for each resource type
  + Maximum resource table for each process
  + Resources allocated to processes
  + Input a resource allocation for all process
  + Calculate deadlock occurs or not
  + If Deadlock – > Output “Deadlock”
  + If No Deadlock – > Output the Process of Execution

Please Implement Bankers algorithm for the given resource allocation.



Find Safe sequence.

**Task 2**

* + - 1. Please check that for process two, is it possible to accept the new request ( 2, 3, 1).
      2. If request is accepted then find the safe sequence if has.