|  |
| --- |
| **S.Y.B.Tech**  **Computer Engineering**  **Lab. : CE 2207 Operating Systems Laboratory (OSL)**  **------------------------------------------------------------------------------------------------**  Assignment No # 7: (Group- ‘A3’)  **Title:** Write a Java program to implement Banker's Algorithm for deadlock handling.  **Objective:** To implement Deadlock handling algorithm.  **Procedure: (Banker’s algorithm)**   1. Accept number of processes in the system (n) 2. Accept number of resources types (m) 3. Accept Maximum Resources i.e. the number of available resources of each type. It is a 1-d array of size m. 4. Accept Max i.e. maximum demand/requirement of each process in a system. It is a 2-d array of size n\*m 5. Accept Allocation i.e. The number of resources of each type currently allocated/granted to each process. It is a 2-d array of size n\*m 6. Calculate Need matrix i.e indicates the remaining resource need of each process. It is a 2-d array of size n\*m.   Need [ i, j ] = Max [ i, j ] – Allocation [ i, j ]   1. Calculate Allocated: Total number of allocated resources of each type. Addition of resource type column - wise from Allocation array. It is a 1-d array of size m 2. Calculate Work/Available/free resources: It is a 1-d array of size m   Work[i]=Available[i] - Allocated[i]   1. Find One Safe Sequence as per the resources needed by process and the free resources. 2. Display the safe sequence.   **(Optional/variation: 1. calculate multiple safe sequences 2.Consider the case of request granted or not granted?)**  **Sample Output:** |
|  |

\*\*\* Menu for Banker's Algorithm\*\*\*

1. Accept details

2.Display details

3.Display safe sequence

0.Exit

Enter choice :1

Enter the number of processes :5

Enter the number of resources :3

Enter the number of instances for each resource

For resource 0 : 10

For resource 1 : 5

For resource 2 : 7

Enter the elements of maximum matrix

For Process P0 : 7 5 3

For Process P1 : 3 2 2

For Process P2 : 9 0 2

For Process P3 : 2 2 2

For Process P4 : 4 3 3

Enter the elements of allocation matrix

For Process P0 : 0 1 0

For Process P1 : 2 0 0

For Process P2 : 3 0 2

For Process P3 : 2 1 1

For Process P4 : 0 0 2

\*\*\* Menu for Banker's Algorithm \*\*\*

1.Accept details

2.Display details

3.Display safe sequence

0.Exit

Enter choice :2

Process Allocation Max Available Need

P0 0 1 0 7 5 3   3 3 2 7 4 3

P1 2 0 0 3 2 2 1 2 2

P2 3 0 2 9 0 2 6 0 0

P3 2 1 1 2 2 2 0 1 1

P4 0 0 2 4 3 3 4 3 1

\*\*\*Menu for Banker's Algorithm\*\*\*

1.Accept details

2.Display details

3.Display safe sequence

0.Exit

Enter choice:3

Safe sequence is:

P1->P3->P4->P0->P2

\*\*\* Menu for Banker's Algorithm \*\*\*

1.Accept details

2.Display details

3.Display safe sequence

0.Exit

Enter choice :0

End of program