CS331

S5 SEMESTER

System Software Lab

Github : [ceccs18c59/cs331: System Software Lab (github.com)]

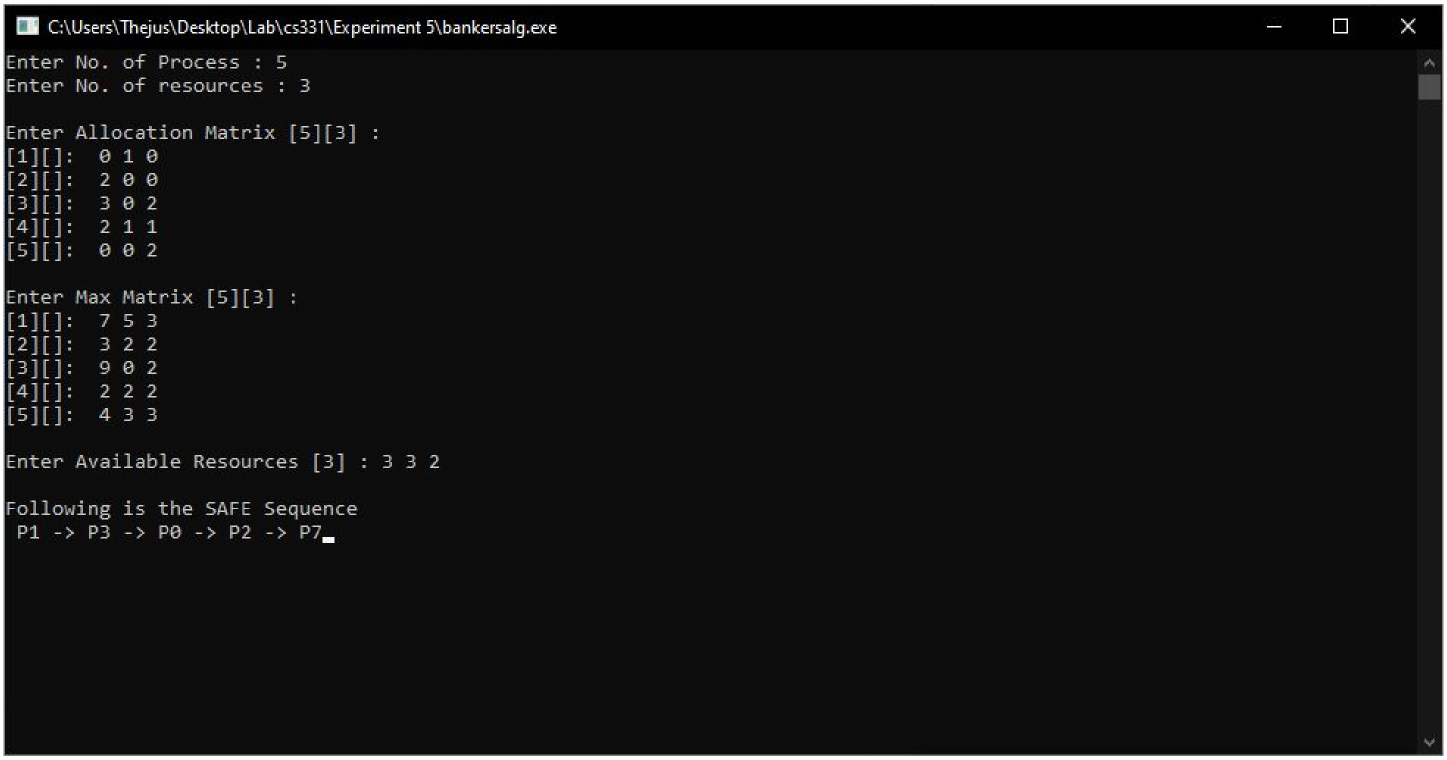
Experiment No 5

Write a C program to simulate the Banker's Algorithm to prevent DeadLocks.

Program

```
#include <stdio.h>
#include <conio.h>

void main()
{
    int processes, resources, i, j, ind;
    int allocation[10][10], max[10][10], available[10];

    printf("Enter No. of Process : ");
    scanf("%d", &processes);
    printf("Enter No. of resources : ");
    scanf("%d", &resources);
    printf("\nEnter Allocation Matrix [%d][%d] : \n", processes, resources);

    for (i = 0; i < processes; i++)
    {
        printf("[%d][]:\t", i + 1);
        for (j = 0; j < resources; j++)
            scanf("%d", &allocation[i][j]);
    }

    printf("\nEnter Max Matrix [%d][%d] : \n", processes, resources);

    for (i = 0; i < processes; i++)
    {
        printf("[%d][]:\t", i + 1);
        for (j = 0; j < resources; j++)
            scanf("%d", &max[i][j]);
    }

    printf("\nEnter Available Resources [%d] : ", resources);

    for (i = 0; i < resources; i++)
        scanf("%d", &available[i]);

    int finish[resources], need[processes][resources], ans[resources];

    for (int k = 0; k < resources; k++)
        finish[k] = 0;
    for (i = 0; i < processes; i++)
        for (j = 0; j < resources; j++)
            need[i][j] = max[i][j] - allocation[i][j];

    for (int k = 0; k < processes; k++)
    {
        for (i = 0; i < processes; i++)
        {
            if (finish[i] == 0)
            {

                int flag = 0;
                for (j = 0; j < resources; j++)
                {
                    if (need[i][j] > available[j])
                    {
                        flag = 1;
                        break;
                    }
                }

                if (flag == 0)
                {
                    ans[ind++] = i;
                    for (int y = 0; y < resources; y++)
                        available[y] += allocation[i][y];
                    finish[i] = 1;
                }
            }
        }
    }

    printf("\nFollowing is the SAFE Sequence\n");
    for (i = 0; i < processes - 1; i++)
        printf(" P%d ->", ans[i]);
    printf(" P%d", ans[processes - 1]);

    getch();
}
```

Output



C:\Users\Thejus\Desktop\Lab\cs331\Experiment 5\bankersalg.exe

```
Enter No. of Process : 5
Enter No. of resources : 3

Enter Allocation Matrix [5][3] :
[1][]:  0 1 0
[2][]:  2 0 0
[3][]:  3 0 2
[4][]:  2 1 1
[5][]:  0 0 2

Enter Max Matrix [5][3] :
[1][]:  7 5 3
[2][]:  3 2 2
[3][]:  9 0 2
[4][]:  2 2 2
[5][]:  4 3 3

Enter Available Resources [3] : 3 3 2

Following is the SAFE Sequence
 P1 -> P3 -> P0 -> P2 -> P7
```