

## OPERATING SYSTEM LAB EXPERIMENTS

### PROGRAM 01 : Bankers Algorithm....

```
#include<iostream>
using namespace std;

int m, n;
int MAX[10][10], ALLOC[10][10], NEED[10][10], AVAI[10];
int isLess(int proc, int WORK[])
{
    for(int i=0; i<m; i++)
        if(NEED[proc][i] > WORK[i])
            return 0;
    return 1;
}
int isLessVector(int a[], int b[], int n)
{
    for(int i=0; i<n; i++)
        if(a[i] > b[i])
            return 0;
    return 1;
}
int visited(int finish[])
{
    int i;
    for(i=0; i<n; i++)
        if(finish[i] == 0)
            break;
    if(i == n)
        return 1;
    return 0;
}
void BankersAlgo()
{
    int finish[10] = {0}, WORK[10], seq[10];

    for(int i=0; i<m; i++) WORK[i] = AVAI[i];
    for(int i=0; i<n; i++)
        for(int j=0; j<m; j++)
            NEED[i][j] = MAX[i][j] - ALLOC[i][j];

    int j = 0;
    int counter = 0;
    while(counter <= 3)
    {
        for(int i=0; i<n; i++)
        {
            if(finish[i] == false && isLess(i, WORK))
            {
                finish[i] = true;
                for(int k=0; k<m; k++) { WORK[k] += ALLOC[i][k]; }
                seq[j++] = i;
            }
        }
        counter++;
        if(visited(finish) == 1)
            break;
    }
}
```

## OPERATING SYSTEM LAB EXPERIMENTs

```
{
    cout << "\nSAFE SEQUENCE : ";
    for(int i=0; i<n; i++)
        cout << "P" << seq[i] << " ";
    return;
}
}
cout << "\nThe system is in DEADLOCK state!!";
}

void resourceReq()
{
    int proc, REQ[10];
    cout << "\n\nRESOURCE - REQUEST\n";
    cout << "\nEnter the requesting process : "; cin >> proc;

    cout << "Enter the REQUEST Vector : ";
    for(int i=0; i<m; i++) { cin >> REQ[i]; }

    if(!isLess(proc, REQ))
    {
        if(isLessVector(REQ, AVAI, m))
        {
            for(int i=0; i<n; i++)
            {
                ALLOC[proc][i] += REQ[i];
                NEED[proc][i] -= REQ[i];
                AVAI[i] -= REQ[i];
            }

            cout << "\nYes! The request will be granted immediately\n";
            BankersAlgo();
            return;
        }
        else
        { cout << "Resources cannot be granted since resources are not
available!"; }
    }
    else
    { cout << "Resources cannot be granted since process has exceeded the
max limit!"; }
}

int main()
{
    cout << "Enter the number of processes : "; cin >> n;
    cout << "Enter the number of resources : "; cin >> m;
    cout << "Enter the MAX Matrix : \n";
    for(int i=0; i<n; i++)
        for(int j=0; j<m; j++)
            cin >> MAX[i][j];

    cout << "\nEnter the ALLOCATION Matrix : \n";
```

## OPERATING SYSTEM LAB EXPERIMENTs

```
for(int i=0; i<n; i++)
    for(int j=0; j<m; j++)
        cin >> ALLOC[i][j];

    cout << "\nEnter the AVAILABLE Vector : ";
    for(int i=0; i<m; i++)
        cin >> AVAI[i];

    BankersAlgo();
    resourceReq();
}
```

### OUTPUT :

```
Enter the number of processes : 5
Enter the number of resources : 3
Enter the MAX Matrix :
7 5 3
3 2 2
9 0 2
2 2 2
4 3 3
```

```
Enter the ALLOCATION Matrix :
0 1 0
2 0 0
3 0 2
2 1 1
0 0 2
```

```
Enter the AVAILABLE Vector : 3 3 2
```

```
SAFE SEQUENCE : P1 P3 P4 P0 P2
```

```
RESOURCE - REQUEST
```

```
Enter the requesting process : 1
```

```
Enter the REQUEST Vector : 1 0 2
```

```
Yes! The request will be granted immediately
```

```
SAFE SEQUENCE : P1 P3 P4 P0 P2
```

```
c:\Users\hayst\Documents\Coding-Programming\Operating Systems (Git)\Experiment05>
```

## OPERATING SYSTEM LAB EXPERIMENTS

```
Enter the number of processes : 5
Enter the number of resources : 4
Enter the MAX Matrix :
0 0 1 2
2 7 5 0
6 6 5 6
4 3 5 6
0 6 5 2

Enter the ALLOCATION Matrix :
0 0 1 2
2 0 0 0
0 0 3 4
2 3 5 4
0 3 3 2

Enter the AVAILABLE Vector : 2 1 0 0

SAFE SEQUENCE : P0 P3 P4 P1 P2

RESOURCE - REQUEST
Enter the requesting process : 2
Enter the REQUEST Vector : 0 1 1 0
Resources cannot be granted since resources are not available!
c:\Users\hayst\Documents\Coding-Programming\Operating Systems (Git)\Experiment05>
```

### **PROGRAM 02 : Deadlock Detection algorithm....**

```
#include<iostream>
using namespace std;

int m, n;
int REQ[10][10], ALLOC[10][10], NEED[10][10], AVAI[10];
int isLess(int proc, int WORK[])
{
    for(int i=0; i<m; i++)
        if(REQ[proc][i] > WORK[i])
            return 0;
    return 1;
}
int isLessVector(int a[], int b[], int n)
{
    for(int i=0; i<n; i++)
        if(a[i] > b[i])
            return 0;
    return 1;
}
int visited(int finish[])
{
    int i;
    for(i=0; i<n; i++)
```

## OPERATING SYSTEM LAB EXPERIMENTS

```
        if(finish[i] == 0)
            break;
    if(i == n)
        return 1;
    return 0;
}
void deadlockDetection()
{
    int j = 0;
    int counter = 0;
    int finish[10] = {0}, WORK[10], seq[10];

    for(int i=0; i<m; i++) WORK[i] = AVAI[i];
    while(counter <= 3)
    {
        int found = 0;
        for(int i=0; i<n; i++)
        {
            if(finish[i] == false && isLess(i, WORK))
            {
                finish[i] = true;
                for(int k=0; k<m; k++) { WORK[k] += ALLOC[i][k]; }
                seq[j++] = i;
                found++;
            }
        }
        if(found == 0) { break; }
        counter++;
        if(visited(finish) == 1)
        {
            cout << "\nThe system is NOT in DEADLOCK STATE!!";
            cout << "\nSAFE SEQUENCE : ";
            for(int i=0; i<n; i++)
                cout << "P" << seq[i] << " ";
            return;
        }
    }
    cout << "\nThe system is in DEADLOCK state!!";
}

int main()
{
    cout << "Enter the number of processes : "; cin >> n;
    cout << "Enter the number of resources : "; cin >> m;
    cout << "Enter the REQUEST Matrix : \n";
    for(int i=0; i<n; i++)
        for(int j=0; j<m; j++)
            cin >> REQ[i][j];

    cout << "\nEnter the ALLOCATION Matrix : \n";
    for(int i=0; i<n; i++)
        for(int j=0; j<m; j++)
            cin >> ALLOC[i][j];
}
```

## OPERATING SYSTEM LAB EXPERIMENTs

```
        cout << "\nEnter the AVAILABLE Vector : ";
        for(int i=0; i<m; i++)
            cin >> AVAI[i];

        deadlockDetection();
    }
```

### OUTPUT :

```
Enter the number of processes : 5
Enter the number of resources : 3
Enter the REQUEST Matrix :
0 0 0
2 0 2
0 0 0
1 0 0
0 0 2
```

```
Enter the ALLOCATION Matrix :
0 1 0
2 0 0
3 0 3
2 1 1
0 0 2
```

Enter the AVAILABLE Vector : 0 0 0

The system is NOT in DEADLOCK STATE!!

SAFE SEQUENCE : P0 P2 P3 P4 P1

c:\Users\hayst\Documents\Coding-Programming\Operating Systems (Git)\Experiment05>

```
Enter the number of processes : 5
Enter the number of resources : 4
Enter the REQUEST Matrix :
0 0 1 2
2 7 5 0
2 0 1 3
4 3 5 6
0 2 3 2
```

```
Enter the ALLOCATION Matrix :
0 0 1 2
2 0 0 0
0 0 3 4
2 3 5 4
0 3 3 2
```

Enter the AVAILABLE Vector : 2 1 3 3

The system is in DEADLOCK state!!

c:\Users\hayst\Documents\Coding-Programming\Operating Systems (Git)\Experiment05>