Resource Request Algorithm

Perumalla Dharan
AP21110010201

Q) Implement Resource Request Algorithm for Deadlock Avoidance:

- **Read Max**
- **Read Allocation**
- **Read Available**
- **Read Request from a process, and check whether the request will be approved or not.**

```
// Initialisation of header files
#include <iostream>
using namespace std;

// Main function
int main()
{
    int n, m;
    cout << "Enter the number of processes: ";
    cin >> n;
    cout << "Enter the number of resources: ";
    cin >> m;

    // Declaring the allocation, maximum, available, need
matrix
    int alloc[n][m], max[n][m], avail[m], need[n][m];

    // Inputting the allocation matrix
    cout << "Enter the allocation matrix: " << endl;
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < m; j++)
        {
```

```
            cin >> alloc[i][j];
        }
    }

    // Inputting the maximum matrix
    cout << "Enter the maximum matrix: " << endl;
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < m; j++)
        {
            cin >> max[i][j];
        }
    }

    // Inputting the available matrix
    cout << "Enter the available matrix: " << endl;
    for (int i = 0; i < m; i++)
    {
        cin >> avail[i];
    }

    // Calculating the need matrix
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < m; j++)
        {
            need[i][j] = max[i][j] - alloc[i][j];
        }
    }

    // Declaring the work and finish array
    int work[m];
    for (int i = 0; i < m; i++)
```

```
    {
        work[i] = avail[i];
    }

    bool finish[n];
    for (int i = 0; i < n; i++)
    {
        finish[i] = false;
    }

    // Declaring the safe sequence array
    int safe_seq[n], count = 0;

    // Inputting the request matrix
    int request[m];
    cout << "Enter the request matrix: " << endl;
    for (int i = 0; i < m; i++)
    {
        cin >> request[i];
    }

    // Checking whether the request will be approved or not
    for (int i = 0; i < m; i++)
    {
        if (request[i] > need[0][i] || request[i] > avail[i])
        {
            cout << "Request cannot be approved" << endl;
            return 0;
        }
    }

    // Updating the matrices
    for (int i = 0; i < m; i++)
```

```
    {
        avail[i] -= request[i];
        alloc[0][i] += request[i];
        need[0][i] -= request[i];
    }

    // Checking for the safe sequence
    while (count < n)
    {
        bool flag = false;
        for (int i = 0; i < n; i++)
        {
            if (finish[i] == false)
            {
                int j;
                for (j = 0; j < m; j++)
                {
                    if (need[i][j] > work[j])
                    {
                        break;
                    }
                }
                if (j == m)
                {
                    for (int k = 0; k < m; k++)
                    {
                        work[k] += alloc[i][k];
                    }
                    safe_seq[count++] = i;
                    finish[i] = true;
                    flag = true;
                }
            }
```

```
        }
        if (flag == false)
        {
            break;
        }
    }

    // Printing the safe sequence
    if (count == n)
    {
        cout << "Safe Sequence: ";
        for (int i = 0; i < n - 1; i++)
        {
            cout << "P" << safe_seq[i] << " -> ";
        }
        cout << "P" << safe_seq[n - 1];
        cout << endl;
    }
    else
    {
        cout << "System is in unsafe state" << endl;
    }
    return 0;
}
```

OUTPUT

```
Enter the number of processes: 5
Enter the number of resources: 3
Enter the allocation matrix:
0 1 0
2 0 0
3 0 2
2 1 1
0 0 2
Enter the maximum matrix:
7 5 3
3 2 2
9 0 2
2 2 2
4 3 3
Enter the available matrix:
3 3 2
Enter the request matrix:
1 0 2
Safe Sequence: P1 -> P3 -> P4 -> P0 -> P2
PS E:\SRM\OS\OS LAB>
```

```
Enter the number of processes: 5
Enter the number of resources: 3
Enter the allocation matrix:
0 1 0
2 0 0
3 0 2
2 1 1
0 0 2
Enter the maximum matrix:
7 5 3
3 2 2
9 0 2
2 2 2
4 3 3
Enter the available matrix:
3 3 2
Enter the request matrix:
9 2 7
Request cannot be approved
PS E:\SRM\OS\OS LAB>
```