Operating System Lab Task Banker’s Algorithm

Name: Ahmed Kasteer

Roll Number: 20F-0336

Section: 4D

#include <iostream>

**using** **namespace** std;

**int** main()

{

    // P0, P1, P2, P3, P4 are the Process names here

**int** n, m, i, j, k;

  n = 5; // Number of processes

  m = 3; // Number of resources

**int** alloc[5][3] = { { 0, 1, 0 }, // P0 // Allocation Matrix

                     { 2, 0, 0 }, // P1

                     { 3, 0, 2 }, // P2

                     { 2, 1, 1 }, // P3

                     { 0, 0, 2 } }; // P4

**int** max[5][3] = { { 7, 5, 3 }, // P0 // MAX Matrix

                   { 3, 2, 2 }, // P1

                   { 9, 0, 2 }, // P2

                   { 2, 2, 2 }, // P3

                   { 4, 3, 3 } }; // P4

**int** avail[3] = { 3, 3, 2 }; // Available Resources

**int** f[n], ans[n], ind = 0;

**for** (k = 0; k < n; k++) {

    f[k] = 0;

  }

**int** need[n][m];

**for** (i = 0; i < n; i++) {

**for** (j = 0; j < m; j++)

      need[i][j] = max[i][j] - alloc[i][j];

  }

**int** y = 0;

**for** (k = 0; k < 5; k++) {

**for** (i = 0; i < n; i++) {

**if** (f[i] == 0) {

**int** flag = 0;

**for** (j = 0; j < m; j++) {

**if** (need[i][j] > avail[j]){

            flag = 1;

**break**;

          }

        }

**if** (flag == 0) {

          ans[ind++] = i;

**for** (y = 0; y < m; y++)

            avail[y] += alloc[i][y];

          f[i] = 1;

        }

      }

    }

  }

**int** flag = 1;

  // To check if sequence is safe or not

**for**(**int** i = 0;i<n;i++)

  {

**if**(f[i]==0)

      {

        flag = 0;

        cout << "The given sequence is not safe";

**break**;

      }

  }

**if**(flag==1)

  {

    cout << "Following is the SAFE Sequence" << endl;

**for** (i = 0; i < n - 1; i++)

        cout << " P" << ans[i] << " ->";

      cout << " P" << ans[n - 1] <<endl;

  }

**return** (0);

}

A picture containing calendar

Description automatically generated

Following is the safe sequence:

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