Name: Lakhan Kumawat

Roll No: 1906055

Branch: CSE-1

Course: CSL4404

Assignment on Banker’s Algorithm

1.    Consider the following example of a system, check whether the system is safe or not, using Banker’s algorithm. Determine the sequence if it is safe.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Process | Allocation  A    B     C | Max  A    B     C | Available  A    B     C | Required  A    B     C |
| P0 | 0     1        0 | 7     5      3 | 3     3      3 |  |
| P1 | 2     0        0 | 3     2      2 |  |  |
| P2 | 3     0        2 | 9     0      2 |  |  |
| P3 | 2     1        1 | 2     2      2 |  |  |
| P4 | 0      0      2 | 4     3      3 |  |  |

2.    Consider the following example of a system, check whether the system is safe or not, using Banker’s algorithm. Determine the sequence if it is safe.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Process | Allocation  A   B   C   D | Max  A    B     C | Available  A    B     C | Required  A    B     C |
| P0 | 0   0    1    2 | 0    0    1    2 | 3     3      3 | 1     5    2   0 |
| P1 | 1   0    0    0 | 1    7    5    0 |  |  |
| P2 | 1   3    5    4 | 2    3    5    6 |  |  |
| P3 | 0   6    3    2 | 0    6    5    2 |  |  |
| P4 | 0   0    1    4 | 0    6    5    6 |  |  |

Program Code: C++ Lakhan Kumawat -1906055

#include <iostream>

using namespace std;

int main()

{

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

int n, m, i, j, k,no;

cout<<"Number of processes :";

cin>>n; // Number of processes

cout<<"Number of resources :";

cin>>m; // Number of resources

cout<<endl;

cout<<"A B C : Allocation Matrix"<<endl;

int alloc[n][m]; // Allocation Matrix

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

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

cin>>no;

alloc[i][j]=no;

}

}

//----Available

int avail[m];

cout<<endl;

cout<<"A B C : Available Resources"<<endl;

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

cin>>no;

avail[j]=no;

}

// MAX Matrix

cout<<endl;

cout<<"A B C : MAX Resources"<<endl;

int MAX[n][m];

int require[n][m]; //Required Resources

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

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

cin>>no;

MAX[i][j]=no;

require[i][j]=MAX[i][j]-alloc[i][j];

}

}

cout<<endl;

int update[n] , result[n] , pno=0; //pno to keep track of every process ---Update to mark if the process is included

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

update[k]=0;

}

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

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

if(update[k]==0){

int flag=0;

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

if(require[k][j]>avail[j]){

flag=1;

break;

}

}

if(flag==0){

result[pno++]=k;

for(int y=0;y<m;y++){

avail[y]+=alloc[k][y];

}

update[k]=1;

}

}

}

}

// Print the RESULTS

int sum=0,p=(n\*(n-1))/2; //Sum of n natural numbers n here is our n-1

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

sum+=result[i];

}

if(sum!=p){

cout<<"Not a Safe sequence :| "<<endl;

}

else{

cout<<"Available Safe Sequence :"<<endl;

for(int h=0;h<n-1;h++){

cout<<"P"<<result[h]<<"-->";

}

cout<<"P"<<result[n-1];

}

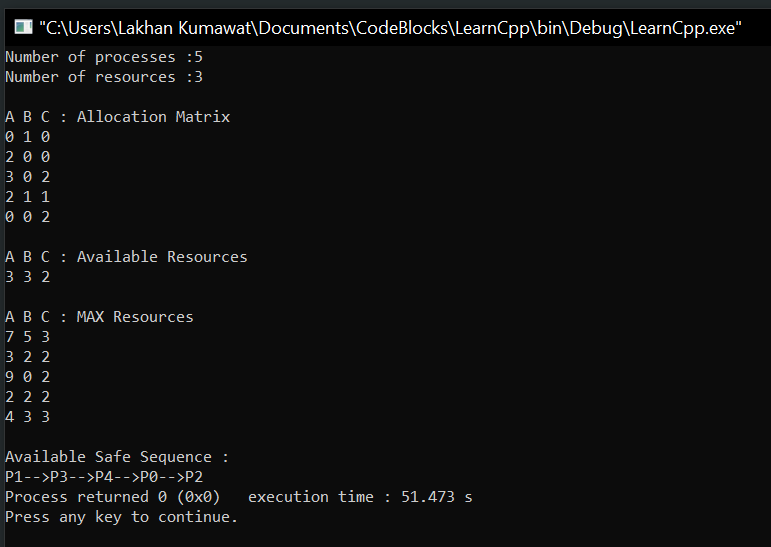
return 0;

}

Lakhan Kumawat -1906055

Outputs: Lakhan Kumawat -1906055

Solution 1:



Solution 2:

