**Institute of Engineering & Management**

**Department of Computer Science & Engineering**

**Operating System Lab for 3rd year 6th semester 2019**

**Code: CS 693**

**Date:** 03/04/19

**WEEK-7**

**Assignment-1**

**Problem Statement:** Implement Banker’s Algorithm

**Source Code:**

#include <iostream>

#include <vector>

bool find\_safe(std::vector<int> avail, std::vector<std::vector<int>> &need, std::vector<std::vector<int>> &alloc, std::vector<int> &safe)

{

int n = avail.size(), m = safe.size();

std::vector<int> done(m,0);

for(int i=0;i<m;i++)

{

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

{

if(done[j] == 1)

continue;

int flag = 0;

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

{

if(avail[k] < need[j][k])

{

flag = 1;

break;

}

}

if(flag == 0)

{

done[j] = 1;

safe[i] = j;

break;

}

}

if(safe[i] == -1)

return false;

else{

for(int j=0;j<n;j++)

avail[j] += alloc[safe[i]][j];

}

}

return true;

}

int main()

{

std::cout<<"\n\t----Banker's Algorithm----\n";

int n, m, allc=0, req;

std::cout<<"Enter no. of resources types: ";

std::cin>>n;

std::cout<<"Enter no. of processes: ";

std::cin>>m;

std::vector<int> r(n,0), avail(n,0), safe(m,-1), req\_v(n,0);

std::vector<std::vector<int>> max(m,std::vector<int>(n,0)), alloc(m,std::vector<int>(n,0)), need(m,std::vector<int>(n,0));

std::cout<<"Enter the total no of instances of each resource type: ";

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

std::cin>>r[i];

std::cout<<"Enter the max need of each process: \n";

for(int i=0;i<m;i++)

for(int j=0;j<n;j++)

std::cin>>max[i][j];

std::cout<<"Enter the allocated no. of resources: \n";

for(int i=0;i<m;i++)

for(int j=0;j<n;j++)

std::cin>>alloc[i][j];

for(int i=0;i<m;i++)

for(int j=0;j<n;j++)

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

std::cout<<"Need Matrix:\n";

for(int i=0;i<m;i++)

{

for(int j=0;j<n;j++)

std::cout<<need[i][j]<<"\t";

std::cout<<"\n";

}

for(int j=0;j<n;j++)

{

allc = 0;

for(int i=0;i<m;i++)

allc += alloc[i][j];

avail[j] = r[j] - allc;

}

std::cout<<"available: ";

for(auto i: avail)

std::cout<<i<<" ";

std::cout<<"\n";

if(find\_safe(avail, need, alloc, safe))

{

std::cout<<"\nSafe State!\nSafe Sequence:";

for(int i=0;i<m;i++)

std::cout<<" p"<<safe[i];

std::cout<<"\n";

}

else

std::cout<<"\nNot safe!\n";

std::cout<<"\nEnter the requesting process no.: ";

std::cin>>req;

std::cout<<"Enter the request vector: ";

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

std::cin>>req\_v[i];

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

alloc[req][i] += req\_v[i];

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

need[req][i] -= req\_v[i];

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

avail[i] -= req\_v[i];

if(find\_safe(avail, need, alloc, safe))

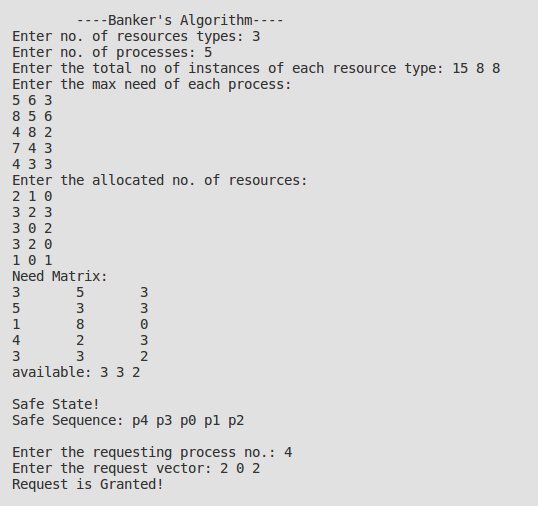
std::cout<<"Request is Granted!\n\n";

else

std::cout<<"Request is Denied!\n\n";

}

**Screen-Shot:**

****