

OPERATING SYSTEM ASSIGNMENT

**Submitted** to:

Mr.Manpreet Singh

**Submitted by:**

Name: Mahendra Aarumalla

Section: K17GB

Admission no : 11703604

Roll no: RK17GBA34

Ques 11. Reena’s operating system uses an algorithm for deadlock avoidance to manage the allocation of resources say three namely A, B, and C to three processes P0, P1, and P2. Consider the following scenario as reference .user must enter the current state of system as given in this example :

Suppose P0 has 0,0,1 instances , P1 is having 3,2,0 instances and P2 occupies 2,1,1 instances of A,B,C resource respectively.

Also the maximum number of instances required for P0 is 8,4,3 and for p1 is 6,2,0 and finally for P2 there are 3,3,3 instances of resources A,B,C respectively. There are 3 instances of resource A, 2 instances of resource B and 2 instances of resource C available. Write a program to check whether Reena’s operating system is in a safe state or not in the following independent requests for additional resources in the

current state:

1. Request1: P0 requests 0 instances of A and 0 instances of B and 2 instances of C.

2. Request2: P1 requests for 2 instances of A, 0 instances of B and 0 instances of C.

All the request must be given by user as input

***Algorithm:***

Step1 : Enter the no of Processes

Step2: Enter the no of Resources

Step3: Enter the Instances of Resources

Step4: Enter the Allocation Instances in a Matrix form

Stept5: Enter the Maximum Instances in a Matrix form

Step6: You will receive the Needed Matrix followed by Processes.

Code:

#include<conio.h>

#include<stdio.h>

int main() {

int n;

int r;

int i;

int j;

int k;

int m;

int mahi;

int avail[10];

int p[10];

int Required[10][10];

int allocation[10][10];

int maximum[10][10];

printf("\nPlease enter the number of process :");

scanf("%d",&n);

printf("\n Please enter no of resources available : ");

scanf("%d",&r);

printf("\nPlease enter insatnces for resources :\n");

for(i=0; i<r; i++) {

printf("R%d ",i+1);

scanf("%d",&avail[i]);

}

printf("\n Please enter allocation matrix \n");

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

p[i]=0;

for(j=0; j<r; j++) {

printf("p%d r%d\t",i+1,j+1);

scanf("%d",&allocation[i][j]);

}

}

printf("\n Please enter maximum matrix \n");

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

for(j=0; j<r; j++) {

printf("p%d r%d\t",i+1,j+1);

scanf("%d",&maximum[i][j]);

}

}

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

printf("\np%d\t",i+1) ;

for(j=0; j<r; j++) {

Required[i][j]=maximum[i][j]-allocation[i][j];

printf("\t%d",Required[i][j]);

}

}

k=0;

mahi=0;

printf("\n\n");

while(k<15) {

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

m=0;

for(j=0; j<r; j++) {

if(p[i]==1) break;

if(Required[i][j]<=avail[j]) {

m++;

}

if(m==r) {

for(j=0; j<r; j++) {

avail[j]+=allocation[i][j];

}

printf("p%d\t",i+1);

p[i]=1;

mahi++;

}

}

}

k++;

}

if(mahi<n-1) {

printf("\nDeadlock ");

}

getch();

}

Output 1:



Output 2:

