



VIT[®]
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

School of Computer Science and Engineering (SCOPE)

Assessment – 4

Operating System (Lab Component)

MCSE504P

Course Name : Operating System (Lab Component)

Course Code : MCSE504P

Course instructor: SALEEM DURAI M.A

Name : NIDHI SINGH

Registration no. : 22MAI0015

Simulation of Bankers algorithm to check whether the given system is in safe state or not. Also check whether addition resource requested can be granted immediately.

Program :-

```
#include<stdio.h>
#include<stdlib.h>
void print(int x[][10],int n,int m){
    int i,j;
    for(i=0;i<n;i++){
        printf("\n");
        for(j=0;j<m;j++){
            printf("%d\t",x[i][j]);
        }
    }
}
//Resource Request algorithm
void res_request(int A[10][10],int N[10][10],int AV[10][10],int pid,int m)
{
    int reqmat[1][10];
    int i;
    printf("\n Enter additional request :- \n");
    for(i=0;i<m;i++){
        printf(" Request for resource %d : ",i+1);
        scanf("%d",&reqmat[0][i]);
    }
    for(i=0;i<m;i++)
        if(reqmat[0][i] > N[pid][i]){
            printf("\n Error encountered.\n");
            exit(0);
        }
    for(i=0;i<m;i++)
        if(reqmat[0][i] > AV[0][i]){
            printf("\n Resources unavailable.\n");
            exit(0);
        }
    for(i=0;i<m;i++){
        AV[0][i]-=reqmat[0][i];
        A[pid][i]+=reqmat[0][i];
        N[pid][i]-=reqmat[0][i];
    }
}
//Safety algorithm
int safety(int A[][10],int N[][10],int AV[1][10],int n,int m,int a[])
{
    int i,j,k,x=0;
    int F[10],W[1][10];
    int pflag=0,flag=0;
    for(i=0;i<n;i++){
        F[i]=0;
        for(i=0;i<m;i++)
            W[0][i]=AV[0][i];
        for(k=0;k<n;k++){
            for(i=0;i<n;i++){
```

```
        if(F[i] == 0){
            flag=0;
            for(j=0;j<m;j++){
                if(N[i][j] > W[0][j])
                    flag=1;
            }
            if(flag == 0 && F[i] == 0){
                for(j=0;j<m;j++)
                    W[0][j]+=A[i][j];
                F[i]=1;
                pflag++;
                a[x++]=i;
            }
        }
    }
    if(pflag == n)
        return 1;
}
return 0;
}

void accept(int A[][10],int N[][10],int M[10][10],int W[1][10],int *n,int *m)
{
    int i,j;
    printf("\n Enter total no. of processes : ");
    scanf("%d",n);
    printf("\n Enter total no. of resources : ");
    scanf("%d",m);
    for(i=0;i<*n;i++){
        printf("\n Process %d\n",i+1);
        for(j=0;j<*m;j++){
            printf(" Allocation for resource %d : ",j+1);
            scanf("%d",&A[i][j]);
        }
    }
    for(i=0;i<*n;i++){
        printf("\n Process %d\n",i+1);
        for(j=0;j<*m;j++){
            printf(" Maximum for resource %d : ",j+1);
            scanf("%d",&M[i][j]);
        }
    }
    printf("\n Available resources : \n");
    for(i=0;i<*m;i++){
        printf(" Resource %d : ",i+1);
        scanf("%d",&W[0][i]);
    }

    for(i=0;i<*n;i++)
        for(j=0;j<*m;j++)
            N[i][j]=M[i][j]-A[i][j];

    printf("\n Allocation Matrix\t\tMaximum Requirement Matrix\t\tNeed Matrix");
    printf("\n\tR1\tR2\t\t\t\tR1\tR2\t\t\t\tR1\tR2\n");
    for(i=0;i<*n;i++){
```

```
        printf("\n");
        for(j=0;j<*m;j++)
            printf("\t%d",A[i][j]);
        printf("\t\t");
        for(j=0;j<*m;j++)
            printf("\t%d",M[i][j]);
        printf("\t\t");
        for(j=0;j<*m;j++){
            printf("\t%d",N[i][j]);
        }
    }
}

int banker(int A[][10],int N[][10],int W[1][10],int n,int m){
    int j,i,a[10];
    j=safety(A,N,W,n,m,a);
    if(j != 0 ){
        printf("\n\n");
        for(i=0;i<n;i++)
            printf(" P%d ",a[i]);
        printf("\n A safety sequence has been detected.\n");
        return 1;
    }else{
        printf("\n Deadlock has occured.\n");
        return 0;
    }
}

int main(){
    int ret;
    int A[10][10];
    int M[10][10];
    int N[10][10];
    int W[1][10];
    int n,m,pid,ch;
    printf("\n DEADLOCK AVOIDANCE USING BANKER'S ALGORITHM\n");
    accept(A,N,M,W,&n,&m);
    ret=banker(A,N,W,n,m);
    if(ret !=0 ){
        printf("\n Do you want make an additional request ? (1=Yes|0=No)");
        scanf("%d",&ch);
        if(ch == 1){
            printf("\n Enter process no. : ");
            scanf("%d",&pid);
            res_request(A,N,W,pid-1,m);
            ret=banker(A,N,W,n,m);
            if(ret == 0 )
                exit(0);
        }
    }else
        exit(0);
    return 0;
}
```

Output:-

Allocation for resource 1 : PS C:\Users\User\Desktop\c program\OS_LAB> ./banker_re

DEADLOCK AVOIDANCE USING BANKER'S ALGORITHM

Enter total no. of processes : 5

Enter total no. of resources : 2

Process 1

Allocation for resource 1 : 1

Allocation for resource 2 : 1

Process 2

Allocation for resource 1 : 2

Allocation for resource 2 : 1

Process 3

Allocation for resource 1 : 1

Allocation for resource 2 : 0

Process 4

Allocation for resource 1 : 2

Allocation for resource 2 : 1

Process 5

Allocation for resource 1 : 3

Allocation for resource 2 : 4

Process 1

Maximum for resource 1 : 4

Maximum for resource 2 : 4

Process 2

Maximum for resource 1 : 3

Maximum for resource 2 : 4

Process 3

Maximum for resource 1 : 3

Maximum for resource 2 : 3

Process 4

Maximum for resource 1 : 3

Maximum for resource 2 : 4

Process 5

Maximum for resource 1 : 4

Maximum for resource 2 : 4

Available resources :

Resource 1 : 4

Resource 2 : 3

Allocation Matrix

R1 R2

1 1

2 1

1 0

2 1

3 4

Maximum Requirement Matrix

R1 R2

4 4

3 4

3 3

3 4

4 4

Need Matrix

R1 R2

3 3

1 3

2 3

1 3

1 0

P0 P1 P2 P3 P4

A safety sequence has been detected.

Do you want make an additional request ? (1=Yes|0=No)1

Enter process no. : 3

Enter additional request :-

Request for resource 1 : 1

Request for resource 2 : 1

2	1	3	4	1	3
1	0	3	3	2	3
2	1	3	4	1	3
3	4	4	4	1	0

P0 P1 P2 P3 P4

A safety sequence has been detected.

Do you want make an additional request ? (1=Yes|0=No)1

Enter process no. : 3

Enter additional request :-

Request for resource 1 : 1

Request for resource 2 : 1

P2 P3 P4 P0 P1

A safety sequence has been detected.

PS C:\Users\User\Desktop\c program\OS_LAB> gcc banker_re.c -o banker_re.exe