

LAB 7

Write a C program to simulate deadlock detection.

16-8-23

Lab - 7

1) Write a C program to simulate deadlock detection

```
#include <stdio.h>
#define MAX process 10
#define MAX-Resources 10
int process, resources;
int allocation [MAX-process][MAX-Resources];
int max-need [MAX-process][MAX-Resources];
int available [MAX-Resources];
int marked [MAX-process];
int finished [MAX-process];

void initialise () {
    printf ("Enter the no. of process: ");
    scanf ("%d", &process);
    printf ("Enter the no. of resources: ");
    scanf ("%d", &resources);
    printf ("Enter the allocation matrix: 'n'");
    for (i=0; i<process; i++)
    {
        for (j=0; j<resources; j++) {
            scanf ("%d", &allocate[i][j]);
        }
    }
    printf ("Enter the max need matrix 'n'");
    for (i=0; i<process; i++)
    {
        for (j=0; j<resources; j++) {
            scanf ("%d", &max[i][j]);
        }
    }
    printf ("Enter the available resources: 'n'");
    for (i=0; i<resources; i++)
    {
        scanf ("%d", &available[i]);
    }
}
```

void detectDeadlock()

{

for ($i=0$; $i < \text{process}$; $i++$)

{

marked[i] = 0;

finished[i] = 0;

}

int marked_count = 0;

while (marked_count < process)

{

if (! finished[i] && ! marked[i])

{

int can_allocate = 1;

for ($j=0$; $j < \text{resources}$; $j++$)

{

if (max_need[i][j] - allocation[i][j] > available[j])

{

can_allocate = 0;

break;

}

if (can_allocate)

{

marked[i] = 1;

marked_count++;

found = 1;

for ($j=0$; $j < \text{resources}$; $j++$)

{

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

}

break;

}

if (! found) {

printf("Deadlock detected ! : \n");

for ($i=0$; $i < \text{process}$; $i++$)


```

4
    if (!finished[i] && !marked[i]) {
        print ("process i.d In ", i);
3
    }
3
return;
3 3
print ("No deadlock detected In ");
3
int main()
{
    initialize();
    detect deadlock();
    return 0;
3
}

```

Output:

Enter the no. of process: 5

Enter the no. of resources: 3

Enter the allocation matrix:

0 1 0

2 0 0

3 0 2

2 1 1

0 0 2

Enter the max need matrix:

5 3

3 2 2

9 0 2

2 2 2

4 3 3

Enter the available resources: 3 3 2

No deadlock detected.

OUTPUT:

```
C:\Users\Admin\Desktop\bm21cs065\deadlock_deec\bin\Debug\deadlock_deec.exe
Enter number of processes and number of resources required
5 3
Enter total number of required resources 5 for each process
7 5 3
3 2 2
9 0 2
2 2 2
4 3 3
Enter number of allocated resources 5 for each process
0 1 0
2 0 0

3 0 2
2 1 1
0 0 2
Enter number of available resources
1 1 1

Deadlock occurred

Process returned 0 (0x0)   execution time : 65.375 s
Press any key to continue.
```

C:\Users\Admin\Desktop\bm21cs065\deadlock_deec\bin\Debug\deadlock_deec.exe

Enter number of processes and number of resources required

5 3

Enter total number of required resources 5 for each process

7 5 3

3 2 2

9 0 2

2 2 2

4 3 3

Enter number of allocated resources 5 for each process

0 1 0

2 0 0

3 0 2

2 1 1

0 0 2

Enter number of available resources

3 3 2

No deadlock

Process returned 0 (0x0) execution time : 86.778 s

Press any key to continue.