

# Sidharth

2K18/MC/114

## Experiment 8

**Aim:** Write a program to implement Banker's Algorithm. The program should either print the safe sequence of execution of given processes (if any exists) or print "There is a deadlock in the system".

**Example:** consider the following snapshot of a system:

Processes	Allocation			Maximum			Available		
	A	B	C	A	B	C	A	B	C
P0	1	1	2	4	3	3	2	1	0
P1	2	1	2	3	2	2			
P2	4	0	1	9	0	2			
P3	0	2	0	7	5	3			
P4	1	1	2	1	1	2			

**Code:**

```
#include <stdio.h>

int current[5][5], maximum_claim[5][5], available[5];
int processes, resources;
int need[5][5];
int i, j, k, counter = 0;
int seq[5];

void isSafe(){
    for (i=0; i<processes; ++i){
        for (int j = 0; j < resources; ++j){
            need[i][j] = maximum_claim[i][j] - current[i][j];
        }
    }
}
```

```

int finish[i];
for (i=0; i<processes; ++i)
    finish[i] = 0;

int work[resources];
for (i=0; i<resources; ++i)
    work[i] = available[i];

while(counter<processes){
    int p;
    int found = 0;
    for (p=0; p<processes; ++p){
        if(finish[p] == 0){
            for (j=0; j<resources; ++j){
                if (need[p][j]>work[j]){
                    break;
                }
            }
            if (j == resources){
                for (k=0; k<resources; ++k){
                    work[k] += current[p][k];
                }
                seq[counter] = p;
                counter +=1;
                finish[p] = 1;
                found = 1;
            }
        }
    }
    if (found == 0){
        printf("\nThere is a deadlock in the system.");
        return;
    }
}
printf("\nSystem is in Safe State. \nSequence : ");
for (i=0; i<processes; ++i){
    printf("P%d  ", seq[i]);
}
printf("\n");

```

```

}

int main(){

    printf("\nEnter number of Processes: ");
    scanf("%d", &processes);
    printf("Enter number of Resources: ");
    scanf("%d", &resources);
    printf("\nEnter available resources:\n");
    for(i=0; i<resources; i++){
        printf("Resource %d: ", i);
        scanf("%d", &available[i]);
    }

    printf("\nEnter Maximum Resources Table:\n");
    for (i=0; i<processes; i++){
        printf("Process %d: ", i);
        for(j = 0; j<resources; j++){
            scanf("%d", &maximum_claim[i][j]);
        }
    }

    printf("\nEnter Allocated Resources Table:\n");
    for (i=0; i<processes; i++){
        printf("Process %d: ", i);
        for(j = 0; j<resources; j++){
            scanf("%d", &current[i][j]);
        }
    }
    isSafe();
    printf("\n");

    return 0;
}

```

## Output:

```
sidharth001@LAPTOP-2SFRN76F: /mnt/c/Users/Sidharth/os
sidharth001@LAPTOP-2SFRN76F: /mnt/c/Users/Sidharth/os$ cd os
sidharth001@LAPTOP-2SFRN76F: /mnt/c/Users/Sidharth/os$ gcc exp8.c && ./a.out

Enter number of Processes: 5
Enter number of Resources: 3

Enter available resources:
Resource 0: 2
Resource 1: 1
Resource 2: 0

Enter Maximum Resources Table:
Process 0: 4 3 3
Process 1: 3 2 2
Process 2: 9 0 2
Process 3: 7 5 3
Process 4: 1 1 2

Enter Allocated Resources Table:
Process 0: 1 1 2
Process 1: 2 1 2
Process 2: 4 0 1
Process 3: 0 2 0
Process 4: 1 1 2

System is in Safe State.
Sequence : P1 P4 P0 P2 P3

sidharth001@LAPTOP-2SFRN76F: /mnt/c/Users/Sidharth/os$
```