OPERITING SYSTEMS LAB 2

BANKER'S ALGORITHM

Team :-

- Adel Mohamed Mahmoud Hashem 18010880
- Mostafa Ahmed Abd EL-Salam EL-Hossary 18011775
- Mahmoud Ahmed Abd EL-Salam Mohamed 18011654

CODE :-

- **THE MAIN FUNCTION HERE IS "BANKER" IT STANDS FOR BANKER'S ALGORITHM :**

```
void Banker(int C[5][3],int M[5][3],int A[1][3],int need[5][3])
{
    Queue p,seq;
    CreateQueue(&p,5);
    CreateQueue(&seq,5);
    for(int i = 0;i<5;i++)
        QueueAdd(&p , i);

    while(!IsEmpty(&p))
    {
        int found = 0,p_size = p.size;
        for(int i = 0 ; i < p_size;i++)
        {
            int process = QueueOut(&p);
            // Can Give Resources To That process
            if(need[process][0] <= A[0][0] && need[process][1] <= A[0][1] && need[process][2] <= A[0][2])
            {
                // Add Take The Resources Back From The Process
                for (int j = 0; j < 3; ++j) {
                    A[0][j] = A[0][j] + C[process][j];
                }
                found = 1;
                QueueAdd(&seq,process); //Add the process to Sequance Queue
            }else //Not Enough Resourses
                QueueAdd(&p,process); //Append it Back to the Queue

        }
        if (!found)
        {
            printf("\n UnSafe State\n");
            return;
        }
    }

    //Print the Sequance
    printf("\n Safe State\n");
    printf("\n possible execution sequence : ");
    QueuePrint(seq);


}
```

- **TO GET THE DATA FROM THE FILE WE NEEDED TO USE SOME FUNCTIONS :**

1- *Loadfile : to load the file we got from the user.*

```
void LoadFile(char *path)
{
getcwd(path,sizeof(path)); //go to the current path
printf("Enter The File Name ==> ");
char filename[50];
scanf("%s",filename);              // get the file name from the user
strcat(path,filename);
}
```

2- *Getdata : to get specific data from the file.*

```
59  void GetData(int n,int c[][3])
60  {
61      int temp = 0,i=0;
62
63      while (i<n)
64      {
65          while(temp < 3)
66              fscanf(f, "%d", &c[i][temp++]);
67          temp = 0;
68          i++;
69      }
70
71  }
```

3- *subtract_matrices : to separate each matrix (Cp,Mp,A).*

```
73  void subtract_matrices(int rows,int col,int out[][col],int mat1[][col],int mat2[][col])
74  {
75      // subtract two matrices
76    for (int i = 0; i < rows; ++i)
77
78      for (int j = 0; j < col; ++j) {
79        out[i][j] = mat1[i][j] - mat2[i][j];
80      }
81
82  }
```

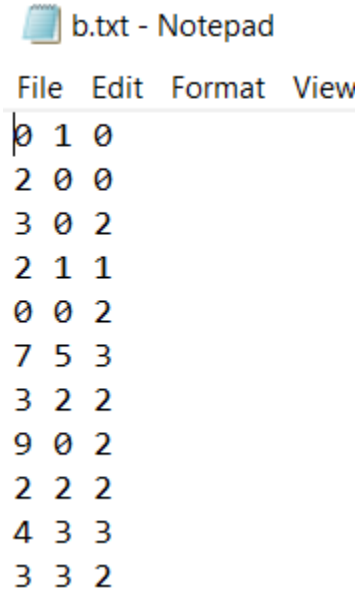
- **WE USED QUEUE DATA STRACT. TO HANDLE THE PROCESSES :**

```
6   //Build Queue
7   typedef struct {
8       int front;
9       int rear;
10      int size;
11      int max;
12      int *data;
13  }Queue;
14
15  void CreateQueue(Queue* q ,int n);
16  int IsEmpty(Queue* q);       //check if queue is empty
17  void QueueAdd(Queue* q , int ndata);
18  int QueueOut(Queue* q);
19  void QueuePrint (Queue q);
```

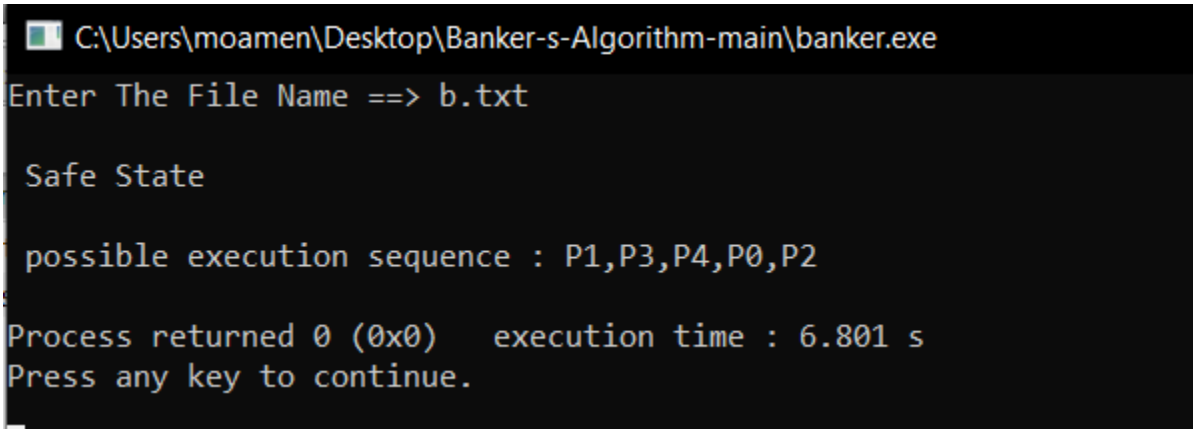
SAMPLE RUNS :-

- **WE TRIED A LOT OF EXAMPLES ,HERE'S TWO OF THEM :-**

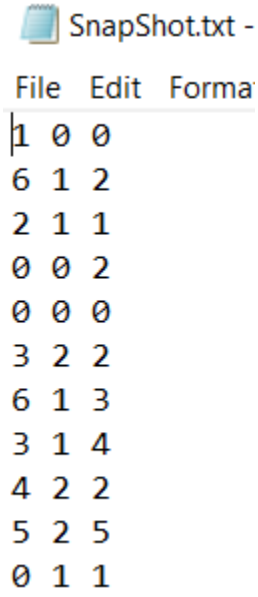
1- *b.txt file*



The run :-



2- *SnapShot.txt file*



The run :-