C:\Users\aswin\Documents\bankalgorithmos.c - Embarcadero Dev-C++ 6.3

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 9.2.0 64-bit Release

(globals)

Project C

bankalgorithmos.c

```c
#include <stdio.h>
#include <conio.h>
int main()
{
int Max[10][10], need[10][10], alloc[10][10], avail[10], completed[10], safeSequence[10];
int p, r, i, j, process, count;
count = 0;
printf("Enter the no of processes : ");
scanf("%d", &p);
for(i = 0; i< p; i++)
    completed[i] = 0;
printf("Enter the no of resources : ");
scanf("%d", &r);
printf("Enter the Max Matrix for each process : \n");
for(i = 0; i < p; i++)
{
    printf("For process %d : ", i + 1);
    for(j = 0; j < r; j++)
        scanf("%d", &Max[i][j]);
}
printf("Enter the allocation for each process : \n");
for(i = 0; i < p; i++)
{
    printf("For process %d : ",i + 1);
    for(j = 0; j < r; j++)
        scanf("%d", &alloc[i][j]);
}
printf("Enter the Available Resources : ");
for(i = 0; i < r; i++)
        scanf("%d", &avail[i]);
```

```
    for(i = 0; i < p; i++)
        for(j = 0; j < r; j++)
            need[i][j] = Max[i][j] - alloc[i][j];
do
{
    printf("\nMax matrix:\tAllocation matrix:\n");
    for(i = 0; i < p; i++)
    {
        for( j = 0; j < r; j++)
            printf("%d  ", Max[i][j]);
        printf("\t\t");
        for( j = 0; j < r; j++)
            printf("%d  ", alloc[i][j]);
        printf("\n");
    }
    process = -1;
    for(i = 0; i < p; i++)
    {
        if(completed[i] == 0)
        {
            process = i ;
            for(j = 0; j < r; j++)
            {
                if(avail[j] < need[i][j])
                {
                    process = -1;
                    break;
                }
            }
        }
```

```
        if(process != -1)
            break;
    }
    if(process != -1)
    {
        printf("\nProcess %d runs to completion!", process + 1);
        safeSequence[count] = process + 1;
        count++;
        for(j = 0; j < r; j++)
        {
            avail[j] += alloc[process][j];
            alloc[process][j] = 0;
            Max[process][j] = 0;
            completed[process] = 1;
        }
    }
}while(count != p && process != -1);
if(count == p)
{
    printf("\nThe system is in a safe state!!\n");
    printf("Safe Sequence : < ");
    for( i = 0; i < p; i++)
            printf("%d  ", safeSequence[i]);
    printf(">\n");
}
else
    printf("\nThe system is in an unsafe state!!");
getch();
}
```

Compiler (1) Resources Compile Log Debug Find Results Console

Line: 80 Col: 51 Sel: 0 Lines: 89 Length: 1968 Insert Done parsing in 0 seconds

```
C:\Users\aswin\Documents\bankalgorithmos.exe
Enter the no of processes : 3
Enter the no of resources : 3
Enter the Max Matrix for each process :
For process 1 : 1 0 3
For process 2 : 0 1 2
For process 3 : 1 2 0
Enter the allocation for each process :
For process 1 : 1 2 1
For process 2 : 2 0 1
For process 3 : 2 2 1
Enter the Available Resources : 0 1 2

Max matrix:     Allocation matrix:
1  0  3                1  2  1
0  1  2                2  0  1
1  2  0                2  2  1

Process 1 runs to completion!
Max matrix:     Allocation matrix:
0  0  0                0  0  0
0  1  2                2  0  1
1  2  0                2  2  1

Process 2 runs to completion!
Max matrix:     Allocation matrix:
0  0  0                0  0  0
0  0  0                0  0  0
1  2  0                2  2  1

Process 3 runs to completion!
The system is in a safe state!!
Safe Sequence : < 1  2  3  >
```