Lab Assignment 13

# Bankers Algorithm

#include <stdio.h>

#include <stdbool.h>

struct nikhil\_process\_info

{

    int max\_1523[10];

    int allocated\_1523[10];

    int need\_1523[10];

};

int no\_of\_process\_1523, no\_of\_resources\_1523;

void input(struct nikhil\_process\_info process\_1523[no\_of\_process\_1523], int available\_1523[no\_of\_resources\_1523])

{

    for (int i = 0; i < no\_of\_process\_1523; i++)

    {

        printf("Enter process\_1523[%d] info\n", i);

        printf("Enter Maximum Need: ");

        for (int j = 0; j < no\_of\_resources\_1523; j++)

            scanf("%d", &process\_1523[i].max\_1523[j]);

        printf("Enter No. of Allocated Resources for this process\_1523: ");

        for (int j = 0; j < no\_of\_resources\_1523; j++)

        {

            scanf("%d", &process\_1523[i].allocated\_1523[j]);

            process\_1523[i].need\_1523[j] = process\_1523[i].max\_1523[j] - process\_1523[i].allocated\_1523[j];

        }

    }

    printf("Enter Available Resources: ");

    for (int i = 0; i < no\_of\_resources\_1523; i++)

    {

        scanf("%d", &available\_1523[i]);

    }

}

void showTheInfo(struct nikhil\_process\_info process\_1523[no\_of\_process\_1523])

{

    printf("\nPID\tMaximum\t\tAllocated\tNeed\n");

    for (int i = 0; i < no\_of\_process\_1523; i++)

    {

        printf("P[%d]\t", i);

        for (int j = 0; j < no\_of\_resources\_1523; j++)

            printf("%d ", process\_1523[i].max\_1523[j]);

        printf("\t\t");

        for (int j = 0; j < no\_of\_resources\_1523; j++)

            printf("%d ", process\_1523[i].allocated\_1523[j]);

        printf("\t\t");

        for (int j = 0; j < no\_of\_resources\_1523; j++)

            printf("%d ", process\_1523[i].need\_1523[j]);

        printf("\n");

    }

}

bool applySafetyAlgo(struct nikhil\_process\_info process\_1523[no\_of\_process\_1523], int available\_1523[no\_of\_resources\_1523], int safeSequence\_1523[no\_of\_process\_1523])

{

    bool finish\_1523[no\_of\_process\_1523];

    int work\_1523[no\_of\_resources\_1523];

    for (int i = 0; i < no\_of\_resources\_1523; i++)

    {

        work\_1523[i] = available\_1523[i];

    }

    for (int i = 0; i < no\_of\_process\_1523; i++)

        finish\_1523[i] = false;

    bool proceed\_1523 = true;

    int k = 0;

    while (proceed\_1523)

    {

        proceed\_1523 = false;

        for (int i = 0; i < no\_of\_process\_1523; i++)

        {

            bool flag\_1523 = true;

            if (finish\_1523[i] == false)

            {

                for (int j = 0; j < no\_of\_resources\_1523; j++)

                {

                    if (process\_1523[i].need\_1523[j] <= work\_1523[j])

                    {

                        continue;

                    }

                    else

                    {

                        flag\_1523 = false;

                        break;

                    }

                }

                if (flag\_1523 == false)

                    continue;

                for (int j = 0; j < no\_of\_resources\_1523; j++)

                    work\_1523[j] = work\_1523[j] + process\_1523[i].allocated\_1523[j];

                finish\_1523[i] = true;

                safeSequence\_1523[k++] = i;

                proceed\_1523 = true;

            }

        }

    }

    int i;

    for (i = 0; i < no\_of\_process\_1523 && finish\_1523[i] == true; i++)

    {

        continue;

    }

    if (i == no\_of\_process\_1523)

        return true;

    else

        return false;

}

bool isSafeState(struct nikhil\_process\_info process\_1523[no\_of\_process\_1523], int available\_1523[no\_of\_resources\_1523], int safeSequence\_1523[no\_of\_process\_1523])

{

    if (applySafetyAlgo(process\_1523, available\_1523, safeSequence\_1523) == true)

        return true;

    return false;

}

int main()

{

    printf("Enter No of Process\n");

    scanf("%d", &no\_of\_process\_1523);

    printf("Enter No of Resource Instances in system\n");

    scanf("%d", &no\_of\_resources\_1523);

    int available\_1523[no\_of\_resources\_1523];

    int safeSequence\_1523[no\_of\_process\_1523];

    struct nikhil\_process\_info process\_1523[no\_of\_process\_1523];

    printf("\*\*\*\*\*\*\*\*\*\*\*Enter details of processes\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

    input(process\_1523, available\_1523);

    showTheInfo(process\_1523);

    if (isSafeState(process\_1523, available\_1523, safeSequence\_1523))

    {

        printf("\nSystem is in SAFE State\n");

        printf("Safe Sequence is: ");

        for (int i = 0; i < no\_of\_process\_1523; i++)

            printf("P[%d] ", safeSequence\_1523[i]);

        printf("1");

    }

    else

        printf("0");

    return 0;

}

Output

