**Expr 9: Banker’s Algorithm**

**Banker’s Algorithm code:**

#include <stdio.h>

#include <stdbool.h>

#define P 5  // Number of processes

#define R 3  // Number of resources

int main() {

    int need[P][R], allocation[P][R], max[P][R], available[R];

    int finish[P] = {0}, safeSequence[P];

    int i, j, k;

    // Example data (can be modified)

    int alloc[P][R] = {

        {0, 1, 0},

        {2, 0, 0},

        {3, 0, 2},

        {2, 1, 1},

        {0, 0, 2}

    };

    int maximum[P][R] = {

        {7, 5, 3},

        {3, 2, 2},

        {9, 0, 2},

        {2, 2, 2},

        {4, 3, 3}

    };

    int avail[R] = {3, 3, 2};

    for (i = 0; i < P; i++) {

        for (j = 0; j < R; j++) {

            allocation[i][j] = alloc[i][j];

            max[i][j] = maximum[i][j];

            need[i][j] = max[i][j] - allocation[i][j];

        }

    }

    for (i = 0; i < R; i++) {

        available[i] = avail[i];

    }

    int count = 0;

    while (count < P) {

        bool found = false;

        for (i = 0; i < P; i++) {

            if (!finish[i]) {

                bool canAllocate = true;

                for (j = 0; j < R; j++) {

                    if (need[i][j] > available[j]) {

                        canAllocate = false;

                        break;

                    }

                }

                if (canAllocate) {

                    for (k = 0; k < R; k++)

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

                    safeSequence[count++] = i;

                    finish[i] = 1;

                    found = true;

                }

            }

        }

        if (!found) {

            printf("System is not in a safe state (deadlock possible).\n");

            return 1;

        }

    }

    printf("System is in a safe state.\nSafe sequence is: ");

    for (i = 0; i < P; i++)

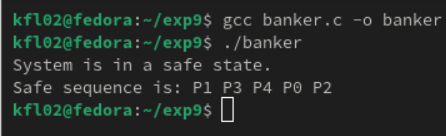
        printf("P%d ", safeSequence[i]);

    printf("\n");

    return 0;

}

**Output:**

****

**Result:**

Thus the Banker Code is implemented in fedora using the C language