#include <stdio.h>

#include <stdlib.h>

#include <semaphore.h>

#include <pthread.h>

int x1 = 1, x2 = 2, x3 = 3, x4 = 4, x5 = 5, x6 = 6;

int w = 0, v = 0, y = 0, z = 0, ans = 0;

sem\_t semAB, semCD, semCDxtra, semEF, semEFxtra, semG, busy;

void\* PAB(void\* mess){

    while (1){

        sem\_wait(&semG);

        sem\_wait(&busy);

        w = x1\*x2;

        v = x3\*x4;

        printf("w = x1\*x2 = %d\n", w);

        printf("v = x3\*x4 = %d\n", v);

        sem\_post(&semCDxtra);

        sem\_post(&semAB);

        sem\_post(&busy);

    }

}

void\* PCD(void\* mess){

    while (1){

        sem\_wait(&semCDxtra);

        sem\_wait(&semAB);

        sem\_wait(&busy);

        y = v\*x5;

        z = v\*x6;

        printf("y = v\*x5 = %d\n", y);

        printf("z = v\*x6 = %d\n", z);

        sem\_post(&semAB);

        sem\_post(&semCD);

        sem\_post(&semEFxtra);

        sem\_post(&busy);

    }

}

void\* PEF(void\* mess){

    while (1){

        sem\_wait(&semEFxtra);

        sem\_wait(&semCD);

        sem\_wait(&semAB);

        sem\_wait(&busy);

        y = w\*y;

        z = w\*z;

        printf("y = w\*y = %d\n", y);

        printf("z = w\*z = %d\n", z);

        sem\_post(&semEF);

        sem\_post(&busy);

    }

}

void\* PG(void\* mess){

    while (1){

        sem\_wait(&semEF);

        sem\_wait(&busy);

        ans = y+z;

        printf("ans = y+z = %d\n", ans);

        sem\_post(&semG);

        sem\_post(&busy);

    }

}

int main(){

    pthread\_t pAB, pCD, pEF, pG;

    sem\_init(&semAB, 0, 0);

    sem\_init(&semCD, 0, 0);

    sem\_init(&semCDxtra, 0, 0);

    sem\_init(&semEF, 0, 0);

    sem\_init(&semEFxtra, 0, 0);

    sem\_init(&semG, 0, 1);

    sem\_init(&busy, 0, 1);

    pthread\_create(&pAB, NULL, &PAB, NULL);

    pthread\_create(&pCD, NULL, &PCD, NULL);

    pthread\_create(&pEF, NULL, &PEF, NULL);

    pthread\_create(&pG, NULL, &PG, NULL);

    while(1){}

    return 0;

}

Các biến semAB, semCD, semEF, đóng vai trò semaphore cho các tiến trình khác thực hiện, các biến semCDxtra, semEFxtra là semaphore cho PCD, PEF thực hiện, biến semG là semaphore cho PAB thực hiện, tạo vòng lặp vô tận. busy đóng vai trò là mutex, khoá và mở vùng tranh chấp.

Nếu các tiến trình khác AB chạy trước:

+ PG bị khoá bởi semEF (do khởi tạo = 0)

+ PEF bị khoá bởi semEFxtra (do khởi tạo = 0)

+ PCD bị khoá bởi semCDxtra (do khởi tạo = 0)

+ PAB được chạy do semG = 1.

+ Sau khi AB chạy xong, CDxtra được kích hoạt để PCD chạy.

+ Sau khi CD chạy xong, EFxtra được kích hoạt để PEF chạy.

+ Sau khi EF chạy xong, semEF được kích hoạt để PG chạy.

+ PG chạy xong, kích hoạt semG để PAB được chạy, tạo thành vòng lặp vô hạn.

Kết quả chạy code:

A screenshot of a computer

Description automatically generated