

Problema “De gatos y ratones”: Tengo k gatos (y desafortunadamente, l ratones) en su casa. Le sirvo la comida a mis gatos en m platos. Gatos y ratones han llegado a un acuerdo para repartirse el tiempo y comida — Pero tienen que convencerme de que están haciendo su trabajo.

Lenguaje: C++11, Entorno: Visual Studio Code

Estrategia de sincronización: La estrategia de sincronización emplea mutexes, condition variables y variables de estado para coordinar la interacción entre gatos y ratones que comparten platos de comida, asegurando que se respeten las reglas del problema (mutex, condition_variable, catSeenMouse, mouseEating).

Código:

```
#include <iostream>

#include <thread>

#include <mutex>

#include <condition_variable>

#include <chrono>


using namespace std;


const int k = 3; // Número de gatos
const int l = 5; // Número de ratones
const int m = 2; // Número de platos


mutex catMutex;
mutex mouseMutex;
condition_variable catCV;
condition_variable mouseCV;
bool catSeenMouse = false;
bool mouseEating = false;


void cat(int id) {
    unique_lock<mutex> catLock(catMutex);
    while (mouseEating) {
        catCV.wait(catLock);
    }

    cout << "Gato " << id << " está comiendo" << endl;

    this_thread::sleep_for(chrono::seconds(2));

    cout << "Gato " << id << " ha terminado de comer" << endl;
```

```
    catSeenMouse = false;

    catLock.unlock();

    mouseCV.notify_all();
}
```

```
void mouse(int id) {
    unique_lock<mutex> mouseLock(mouseMutex);
    if (catSeenMouse) {
        return;
    }
    mouseEating = true;
    cout << "Ratón " << id << " está comiendo" << endl;
    this_thread::sleep_for(chrono::seconds(2));
    cout << "Ratón " << id << " ha terminado de comer" << endl;
    mouseEating = false;
    catCV.notify_all();
}
```

```
int main() {
    thread catThreads[k];
    thread mouseThreads[l];

    for (int i = 0; i < k; i++) {
        catThreads[i] = thread(cat, i + 1);
    }

    for (int i = 0; i < l; i++) {
        mouseThreads[i] = thread(mouse, i + 1);
    }
}
```

```
    for (int i = 0; i < k; i++) {  
        catThreads[i].join();  
    }  
  
    for (int i = 0; i < l; i++) {  
        mouseThreads[i].join();  
    }  
  
    return 0;  
}
```