Instituto Federal da Bahia
Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas
Disciplina: INF009 – Sistemas Operacionais

Vamos estudar o problema do **contador compartilhado**.

Qual é a ideia?

Imagine que existem várias threads tentando *incrementar* o mesmo contador. Existem dois tipos de implementação. Vamos ver o primeiro. Seu papel é entender este código, executar e descrever o que acontece.

Aqui vai o exemplo da implementação em C usando ***pthreads***.

O Programa em C

```c
#include <stdio.h>
#include <stdlib.h>
#include <pthread.h>

// --- Configurações ---
#define NUM_THREADS 5          // Número de threads que vamos criar
#define INCREMENTS_PER_THREAD 100000 // Quantas vezes cada thread vai incrementar o contador

// --- Variáveis Globais Compartilhadas ---
volatile int shared_counter = 0; // O contador compartilhado. 'volatile' sugere ao compilador
                                 // que o valor pode mudar de formas inesperadas (ex: por outra thread)
pthread_mutex_t counter_mutex;   // O mutex para proteger o contador

// --- Função que cada thread executará (sem proteção) ---
void* increment_counter_unsafe(void* arg) {
    long thread_id = (long)arg; // Apenas para identificar a thread, se necessário
    // printf("Thread %ld iniciada (unsafe).\n", thread_id);

    for (int i = 0; i < INCREMENTS_PER_THREAD; i++) {
        // Operação NÃO SEGURA (race condition pode ocorrer aqui)
        // Isso pode ser decomposto em:
```

```
        // 1. Ler shared_counter para um registrador
        // 2. Incrementar o valor no registrador
        // 3. Escrever o valor do registrador de volta para
shared_counter
        // Outra thread pode interferir entre esses passos!
        shared_counter++;
    }
    // printf("Thread %ld finalizada (unsafe).\n", thread_id);
    return NULL;
}
```