

BÀI 15: LẬP TRÌNH ĐA LUỒNG

Multithreading & Synchronization

Mục tiêu bài học

1. **Thread vs Process:** Phân biệt Luồng và Tiến trình.
2. **POSIX Threads (`pthread`):** Tạo, chạy và hủy luồng.
3. **Race Condition:** Hiểu lỗi tranh chấp tài nguyên kinh điển.
4. **Mutex:** Cơ chế khóa để bảo vệ dữ liệu chung.

1. Thread vs Process

Đặc điểm	Process	Thread
Bộ nhớ	Riêng biệt (An toàn).	Chia sẻ chung (Global variable, Heap).
Tốc độ tạo	Chậm (Copy cả bộ nhớ).	Nhanh.
Giao tiếp	Khó (IPC).	Dễ (Độc biến chung).
Rủi ro	1 process chết, process khác vẫn sống.	1 thread lỗi, cả chương trình chết.

2. Tạo Thread (`pthread_create`)

```
#include <pthread.h>

void *my_thread_func(void *arg) {
    printf("Hello from Thread!\n");
    return NULL;
}

int main() {
    pthread_t tid; // Tạo thread
    pthread_create(&tid, NULL, my_thread_func, NULL);
    // Chờ thread kết thúc
    pthread_join(tid, NULL);
    return 0;
}
```

“ **Lưu ý biên dịch:** Phải thêm cờ `-lpthread` (Link pthread library).
`gcc main.c -o app -lpthread` ”

3. Race Condition (Cuộc đua nguy hiểm)

Khi 2 thread cùng ghi vào một biến `count` cùng lúc → Giá trị bị sai.

Ví dụ:

- Thread A đọc `count` (0).
- Thread B đọc `count` (0).
- Thread A tăng lên 1, ghi vào RAM.
- Thread B tăng lên 1, ghi vào RAM (đè lên A).
- **Kết quả:** `count` = 1 (Lẽ ra phải là 2).

4. Mutex (Mutual Exclusion)

Là "cái khóa". Chỉ ai giữ chìa khóa mới được vào nhà vệ sinh (vùng nhớ chung).

```
pthread_mutex_t lock;  
// Khởi tạo  
pthread_mutex_init(&lock, NULL);  
  
// Trong Thread:  
pthread_mutex_lock(&lock); // --- KHÓA ---  
count++; // Vùng an toàn (Critical Section)  
pthread_mutex_unlock(&lock); // --- MỞ ---  
  
// Hủy  
pthread_mutex_destroy(&lock);
```

PHẦN THỰC HÀNH (LAB 15)

Đa luồng đếm số

Yêu cầu

1. Khai báo biến toàn cục `int counter = 0;`.
2. Tạo 2 thread:
 - Thread 1: Tăng `counter` lên 1 triệu lần.
 - Thread 2: Tăng `counter` lên 1 triệu lần.
3. In kết quả cuối cùng.
 - Nếu không dùng Mutex: Kết quả $< 2.000.000$.
 - Nếu dùng Mutex: Kết quả $= 2.000.000$.

Q & A

Hẹn gặp lại ở Bài 16: Giao tiếp IPC!