

BÁO CÁO LINUX THREAD

Bài tập 1

1. Giải thích vai trò và các tham số chính của hàm `pthread_create()` và `pthread_join()`.

- `int pthread_create(pthread_t *thread, // ID của thread mới
const pthread_attr_t *attr, // Thuộc tính thread (NULL = mặc định)
void *(*start_routine)(void *), // Hàm mà thread sẽ chạy
void *arg // Tham số truyền vào hàm trên
);`
- `int pthread_join(pthread_t thread, void **retval);`
 - + `thread`: ID của thread
 - + `retval`: Con trỏ dùng để nhận giá trị trả về từ `pthread_exit` hoặc return của thread (truyền NULL nếu không cần)

2. Một luồng kết thúc khi nào?

Một luồng kết thúc khi:

- return trong hàm thread:
 - + Khi hàm thread return, thread sẽ kết thúc và giá trị của return được gửi đến `pthread_join()`.
 - + Nhưng khi dùng return ở main thread thì nó sẽ kết thúc cả process, kéo theo tất cả các thread bị hủy
- Dùng `pthread_exit()`:
 - + Kết thúc thread nào gọi nó.
 - + Main Thread mà dùng cái này thì mấy thread khác vẫn chạy bình thường.

Bài tập 2

1. Tại sao cần phải sử dụng mutex trong bài toán này?

- Mutex đảm bảo chỉ có 1 thread truy cập tài nguyên chung tại 1 thời điểm.
=> Đảm bảo chương trình chạy đúng.

2. Điều gì sẽ xảy ra nếu chúng ta bỏ qua việc sử dụng mutex? Giải thích tại sao kết quả cuối cùng có thể không chính xác và không ổn định.

- Nếu không sử dụng mutex thì các thread sẽ tranh chấp tài nguyên là counter dẫn đến kết quả counter không chính xác và không ổn định.
- Bởi vì: 3 thread đều tranh chấp 1 tài nguyên là biến counter, khi 1 thread đang tăng counter, chưa kịp tăng thì thread khác nhảy vào lấy counter đi xài => Nên dẫn đến giá trị counter không ổn định.

