

Thực hành NLHDH buổi 3

Một số hàm thao tác với luồng và semaphore

Tạo 1 luồng	<code>int pthread_create (pthread_t *thread_id, const pthread_attr_t *attr, void* (*thread) (void *), void *arg);</code>	<code>thread_id</code> – số hiệu của luồng <code>attr</code> – đặt thuộc tính luồng, NULL mặc định <code>thread</code> – hàm được khởi chạy khi luồng tạo ra <code>arg</code> – tham số cho hàm khởi tạo luồng
Chờ đồng bộ luồng con rồi mới kết thúc	<code>int pthread_join(pthread_t thread_id, void **ret_value);</code>	<code>thread_id</code> – số hiệu của luồng con <code>ret_value</code> – giá trị trả lại của luồng con qua <code>pthread_exit()</code>
Khai báo biến semaphore	<code>sem_t sem;</code>	Khai báo biến semaphore <code>sem</code>
Khởi tạo giá trị	<code>int sem_init (sem_t *sem, int pshared, unsigned int value);</code>	<code>sem</code> – biến semaphore <code>pshared</code> – nếu =0 thì sem chỉ dùng trong nội bộ tiến trình, nếu <>0 thì chia sẻ giữa các tiến trình (phải khai báo dạng shared memory) <code>value</code> – giá trị khởi tạo
Up	<code>int sem_post(sem_t *sem);</code>	<code>sem</code> – biến semaphore
Down	<code>int sem_wait(sem_t *sem);</code>	<code>sem</code> – biến semaphore

Bài 1: Sửa chương trình `guiruttien_peterson.c` để giải quyết bài toán gửi và rút tiền sử dụng giải pháp Peterson.

Bài 2: Viết những phần còn thiếu trong `producers_consumers.c` để giải quyết bài toán Nhà sản xuất – Người tiêu dùng, sử dụng luồng và semaphore.

Bài 3: Biên dịch và chạy chương trình `producers_consumers_process.c`

Bài 3: Biên dịch và chạy chương trình `producer.c`, `consumer.c` và chạy từ 2 cửa sổ riêng biệt