Concurrencia y Paralelismo

Grado en Informática 2025

Práctica 2 – Librerías mutex / semáforos

Ejercicio 1 (**Mutex recursivos**) En la práctica 1 un thread podía interbloquarse a sí mismo cuando intentaba intercambiar una posición con la misma. Una forma de solucionarlo es mediante el uso de mutex recursivos, que permiten al thread que ha bloqueado un mutex bloquearlo un número arbitrario de veces. El mutex solo queda desbloqueado cuando se ha hecho el mismo número de unlocks que de locks.

Es posible implementar mutex recursivos utilizando como base los de la librería pthread, más la función `pthread_t pthread_self()`, que devuelve el identificador del thread que la llama. Implemente estos mútex en `rec_mutex.c`, siguiendo la interfaz definida en `rec_mutex.h`.

Compruebe que funcionan correctamente modificando el apartado 2 de la primera práctica para que no compruebe si se genera la misma posición, pero utilice la librería de mútex recursivos para solventar el problema. Incluya esa versión de swap.c junto con todos los ficheros necesarios para compilarla en la carpeta `rec_mutex`. El programa debe compilar con make.

Ejercicio 2 (**Mutex Read/Write**) Una forma de resolver el problema de lectores/escritores es implementar un mutex donde se puedan hacer bloqueos para lectura (`mutex_read_lock`) y para escritura (`mutex_write_lock`). Un bloqueo de escritura obtiene acceso exclusivo a la zona compartida, pero un bloqueo de lectura es compatible con otros bloqueos de lectura.

Implemente una librería de mutex con bloqueos de lectura/escritura según la intefaz definida en `rw_mutex.h`.

Ejercicio 3 (**Semáforos**) Vamos a implementar, utilizando los mutex y condiciones de la librería pthread, nuestra propia librería de semáforos. Para ello, implemente la interfaz definida en el fichero `sem.h` dentro del fichero `sem.c`. En la estructura `sem_t` solo puede usarse tipos básicos de C, mutex y condiciones.

Implemente el problema del barbero usando semáforos descrito en clase utilizando la librería que acabamos de definir en un fichero `barber.c`. Tanto los barberos como los clientes deberían imprimir un mensaje cuando vayan a cortar el pelo indicando su número de barbero / cliente. El programa con el barbero debería compilar con la orden make.

Entrega

La fecha límite de entrega es el 2 de marzo. Para crear el respositorio y acceder al código inicial, acceda a github classroom en https://classroom.github.com/a/BiZXjDTR. El nombre de los grupos de prácticas debe ser el login de ambas personas, separados por un guión.