Sistemas en Tiempo Real – P1

Programación con hebras POSIX

Buffer acotado

Se trata de terminar la implementación del paquete de buffer acotado de números enteros presentado en la documentación de teoría. Recordar que los mutex y las variables de condición deben inicializarse antes de usarse y destruirse cuando dejan de usarse.

El objeto tendrá las siguientes funciones:

- *void limited_buffer_create (limited_buffer_t *buffer, int buff_size)* → Inicializa la estructura del buffer. El parámetro *buff_size* establecerá el tamaño que va a utilizar el buffer.
- *void limited_buffer_destroy (limited_buffer_t *buffer)* → Liberará el mutex y las variables de condición.
- *void limited_buffer_put (limited_buffer_t *buffer, int elem)* → Añadirá *elem* tras el último elemento insertado en el buffer. Se debe comprobar que el buffer no esté lleno y, al tratarse de un buffer circular, hay que controlar si se ha sobrepasado el límite para volver al principio.
- *void limited_buffer_get (limited_buffer_t *buffer, int *elem)* → Obtendrá el primer elemento del buffer y lo devolverá en *elem* (al ser *elem* un puntero, para acceder a su contenido se debe usar **elem*). Se debe comprobar que el buffer no esté vacío y, al tratarse de un buffer circular, hay que controlar si se ha sobrepasado el límite para volver al principio.

Se deberá crear un fichero .h con las cabeceras, un .c con el código del paquete del buffer y otro .c con el programa principal, que deberá crear 5 productores y 5 consumidores, cada uno de los cuales añadirá/extraerá 50 valores enteros al/del buffer. Mostrar mensajes informativos de los sucesos que se crean importantes.

Deberá garantizarse en todo momento la exclusión mutua, así como la garantía de no sacar elementos de un buffer vacío ni añadirlos a un buffer lleno.

Los archivos de la práctica se comprimirán en un único archivo que será entregado a través del campus virtual usando la tarea creada para tal efecto.