

SCD

Practica 1- Problema del productor-consumidor

Descripción variables

Variables de Buffer

En mi solución del problema sólo ha sido necesario añadir 3 variables de control del buffer donde se almacenarán los datos producidos para su posterior consumición.

La primera es el buffer en sí mismo. Es un array de int de tamaño fijo pero igualado a una constante definida al comienzo de la ejecución. Por lo que se puede modificar simplemente cambiando la variable tam_vec(tamaño del vector) y el programa continuará funcionando correctamente.

La segunda es la variable primera_libre, inicializada a 0. Esta variable será siempre el límite derecho del buffer por así decirlo, nos marcará la siguiente posición en la que almacenar el nuevo dato producido. Puesto que se está planteando una solución FIFO, esta variable de control retorna a la posición 0 (cola circular) cuando alcanza el límite de tamaño del buffer.

La tercera es la variable primera_ocupada, inicializada a 0. Esta variable tiene la función análoga a la primera_libre pero se encarga de avisar al consumidor de cual es el producto más antiguo disponible para consumir. De igual manera que primera_libre retorna a 0 cada vez que alcanza el límite del vector.

Evidentemente, este sistema únicamente funciona sin producir interferencias ni pérdida de datos gracias a los semáforos, que nos garantizan que nunca se consumirá el elemento de primera_ocupada si antes no ha sido producido.

Descripción semáforos

Solución FIFO

Para la solución propuesta con una cola de sistema FIFO sólo son necesarios dos semáforos.

Semaphore libres = tam_vec

Este semáforo llevará la cuenta de cuantas casillas libres tiene el buffer, manteniendo en espera al productor cuando no haya huecos libres en los que guardar el valor producido. Cada vez que se produzca un valor nuevo y se almacene en el buffer será porque previamente hemos pasado por un sem_wait(libres) y el semáforo no se encontraba a 0. Cada vez que se consuma un objeto y por tanto se libere una posición del buffer se realizará un sem_signal por parte del consumidor para avisar al productor de dicho hueco liberado, aumentando el valor del semáforo en 1.

Semaphore llenas = 0

Este otro lleva la cuenta de las casillas ocupadas por valores producidos. Si el semáforo vale 0 (no hay elementos) el consumidor queda en espera en un sem_wait(llenas) hasta que el productor incluya un valor nuevo al buffer y realice el sem_signal correspondiente para aumentar el valor del semáforo en 1. El valor máximo de este semáforo es tam_vector, ya que cada vez que se llena una casilla del buffer el semáforo libres pierde 1. Por tanto cuándo llenas=tam_vec, libres estará a 0 y no se producirán nuevos elementos hasta que no se libere una casilla.

SCD

Practica 1- Problema de los fumadores

Descripción semáforos

-Semaphore mostrador_vacio=1

Este es el semáforo más sencillo de la solución propuesta. Consiste en un semáforo que utilizan todas las hebras y que indica si hay un elemento en el mostrador (valor=0) o si esta vacío (valor=1). Evidentemente comienza inicializado a 1 ya que el mostrador empieza vacío. Cada vez que el estanquero produzca un elemento pasará por un wait para comprobar que el mostrador está vacío y en dicho momento el valor será 0 hasta que se realice un sem_signal por parte del fumador que corresponda. Sólo el estanquero realiza wait sobre este semáforo y sólo los fumadores realizan signal.

-Semaphore fumador[3]

Se trata de un array de 3 semáforos inicializados a 0 que impiden que los fumadores fumen hasta que el estanquero no produzca el ingrediente que necesitan. Una vez producido se realiza un sem_signal sólo al semáforo correspondiente con dicho ingrediente. El fumador que se encuentra en un wait siempre que no esté fumando ya recibirá dicho signal para empezar a fumar volviendo a bloquearse tras dicha acción hasta el siguiente signal.

Es decir sólo la hebra_fumador(i) realizará wait al semáforo fumador[i] y sólo la hebra del estanquero realizará los signals a los 3 semáforos de fumador

Jorge Gangoso Klock 49398653N