Sistemas Operativos II [G4012227] [2022/2023]

Práctica 3 - Sincronización de procesos con mutexes

Resumen

El objetivo de esta práctica es el de conocer el funcionamiento básico de los mecanismos de solución al problema de las carreras críticas usando mutexes y variables de condición, además de desarrollar habilidades en su manejo.

1. Descripción de la práctica

- Programar una versión **modificada** del problema del **productor-consumidor** usando **hilos**, **mutexes** y **variables de condición**.
- El programa debe configurarse para números arbitrarios de productores y consumidores. Para ello, deberemos definir el número de productores P y consumidores C como una constante.
- A diferencia de la versión del problema del productor-consumidor vista en clase de teoría, en esta versión, tanto los productores como los consumidores tendrán que realizar **dos** tareas:
  - Los productores:
    - Producir items (tarea principal).
    - Contribuir a calcular la suma de todos los valores de las posiciones pares de un determinado array.
  - Los consumidores:
    - Consumir items (tarea principal).
    - Contribuir a calcular la suma de todos los valores de las posiciones impares de un determinado array.
- Cada productor P debe producir 18 items.
- Se debe asegurar que, globalmente, **todos los items producidos sean consumidos**.
- El **buffer** de items (**enteros**) debe funcionar como una cola **LIFO** (Last In First Out), de manera que el último ítem en entrar sea el primero en salir.
- El **tamaño del buffer** debe ser **N=12**, definido como una constante en el código.
- Recuerda que el código de la figura 2.32 estudiado en clase considera un buffer de tamaño 1, y por tanto hay que adaptarlo convenientemente.
- Se deberá crear un array T (de **enteros**) de tamaño (110 * (P+C)), donde T[i] = i/2 (división entera).

- Cada vez que un productor termine de incluir un nuevo ítem en el buffer o haya alcanzado el número máximo de items a producir, deberá dedicar parte de su tiempo a **intentar contribuir al sumatorio de los valores de las posiciones pares de T**. A continuación, deberá continuar con su tarea de producir items.
- Del mismo modo, cada vez que un consumidor termine de consumir un ítem del buffer o no haya items que consumir, es decir, todos los productores han terminado de producir y el buffer está vacío, deberá dedicar parte de su tiempo a **intentar contribuir al sumatorio de los valores de las posiciones impares de T**. A continuación, el consumidor deberá continuar con su tarea de consumir items.
- El número de elementos de T que un productor y un consumidor pueden **sumar de cada vez** será un valor aleatorio entre **2 y 4**.
- Tanto los productores como los consumidores **deben imprimir** el valor final del sumatorio al que han contribuido antes de finalizar el programa.
- A los códigos de los productores y los consumidores **deben añadirse llamadas a funciones sleep** en las siguientes fases del código:
  - En el caso de los productores:
    - En la producción de un ítem.
    - En la inclusión en el buffer de un ítem.
    - En la contribución al sumatorio de T.
  - En el caso de los consumidores:
    - En la extracción de un ítem del buffer.
    - En la consumición de un ítem.
    - En la contribución al sumatorio de T.
- Los valores, de tiempo de espera en **segundos**, asociados a cada una de las llamada sleep indicadas en el punto anterior deben ser **parámetros del programa**. Se deben buscar los valores adecuados para que se den los siguientes casos:
  - Los productores terminen su tarea principal antes que el sumatorio de los elementos pares de T y viceversa.
  - Los consumidores terminen su tarea principal antes que el sumatorio de los elementos impares de T y viceversa.
  - Que un productor tenga menos trabajo que un consumidor y viceversa.

2. Práctica voluntaria

La práctica voluntaria sirve para completar tus habilidades sobre sincronización de procesos.

- Resuelve el problema con mutexes, pero **evitando el uso de variables de condición**. Pueden entregarse diversas soluciones.

3. Formato y fecha de entrega

- Hacer un breve **informe (máximo 2 páginas)** incluyendo comentarios y las conclusiones obtenidas para el apartado 1. Sube al Campus Virtual un **único archivo comprimido llamado Apellido1_-Apellido2-Apellido1_Apellido2.zip** incluyendo el código con comentarios exhaustivos sobre su funcionamiento/compilación y el informe.
- En la nota obtenida se tendrá en cuenta la estructura y claridad tanto del informe como del código.
- La evaluación de esta práctica puede estar sujeta a una revisión individual.
- Fecha de entrega: Hasta las 23:59 del día anterior a la primera sesión de la práctica 4 que te corresponda (ver entrega en el Campus Virtual).

4. Cabecera básica con includes y constantes

Usa la siguiente cabecera como base para resolver el ejercicio. Puedes añadir otros include y definir las constantes que consideres necesario.

```
#include <stdlib.h>
#include <stdio.h>
#include <unistd.h>
#include <pthread.h>
#define P 5 // productores
#define C 4 // consumidores
```