

Licenciatura en Computación
PROGRAMACIÓN CONCURRENTE
**Práctica 4: Semáforos POSIX y el patrón
productor-consumidor**

Trimestre 23I

1. Objetivos

- Conocer y utilizar los semáforos POSIX sin nombrar para sincronizar los hilos de un programa concurrente.
- Familiarizarse con el patrón productor-consumidor
- Practicar las arquitecturas prethreading y pipeline.

2. Introducción: revisar el estandar POSIX

(<http://pubs.opengroup.org/onlinepubs/9699919799/nframe.html>) referente a los llamados:

- `int sem_init(sem_t *sem, int pshared, unsigned value);`
- `int sem_wait(sem_t *sem);`
- `int sem_post(sem_t *sem);`

3. Actividades

- Copie, compile y ejecute los programas ejemplo hasta comprender los llamados al sistema para la creación, inicialización y uso de los semáforos POSIX no nombrados.
- El programa ejemplo ProdCons.c funciona correctamente para cuando se tienen varios productores y varios consumidores pero imposibilita la actividad simultánea de un productor y un consumidor sobre el buffer y los índices. Note, sin embargo, que un productor y un consumidor nunca van a trabajar sobre las mismas celdas de memoria pues incrementan índices diferentes y mientras que un productor trabaja sobre una celda vacía, un consumidor trabaja siempre sobre una celda llena. Proponga una solución para permitir el actuar simultáneo de un productor y un consumidor pero que evite que dos productores/consumidores trabajen de manera simultánea sobre la misma celda del buffer y el índice correspondiente. Su programa final debe trabajar con NP productores y NC consumidores, constantes enteras mayores a 1.
- Usando como base el programa ProdCons.c construya un servidor con un pool de hilos (arquitectura prethreading) que simule recibir peticiones de viajes y las coloque en un buffer para que sean atendidas por un pool de hilos que simulan ser taxis. Cada petición de viaje tiene un identificador único que es un entero positivo y una distancia a recorrer entre 1 y 100kms. Un hilo taxi saca la siguiente petición en turno, la despliega en pantalla precedida de su número de hilo, duerme un número

de segundos que es $\lfloor 1/10 \rfloor$ de la distancia y cobra un costo que es $10 \cdot \text{distancia}$. Cada hilo lleva la cuenta de la cantidad de viajes atendidos y la cantidad de dinero generado. Al final, registra estos datos en la memoria compartida. Para terminar, el hilo generador de viajes pone en el buffer un viaje con $\text{id}=-1$ para cada hilo taxi. Cuando un hilo taxi saca del buffer un viaje con $\text{id}=-1$ sabe que ya es momento de registrar su balance y terminar. Cuando todos los hilos taxi han terminado, el hilo generador, antes de terminar, despliega en pantalla: la cantidad de viajes atendidos en total, el id del hilo taxi que más viajes atendió y el id del hilo taxi que más ingreso generó.

- D) Usando como base el programa `ProductorConsumidor.c` elabore un programa concurrente multihilado que tenga una arquitectura de pipeline de longitud 3. El primer hilo lee de pantalla números enteros y los coloca en un buffer para ser atendidos por el segundo hilo. El segundo hilo calcula la secuencia de Collatz para cada número que reciba y coloca el par (número, longitud de la secuencia de Collatz) en un segundo buffer. El tercer hilo saca los pares (número, longitud de la secuencia de Collatz) del segundo buffer y, al vuelo, va calculando cuál de los números procesados tiene la secuencia de Collatz más larga. Para terminar el usuario dará -1 como entrada, cuando el segundo hilo saque ese número sabe que ya debe terminar pero antes debe depositar el par (-1,-1) en el segundo buffer. Cuando el tercer hilo saca el par (-1,-1) despliega la longitud de la secuencia de Collatz más larga y el número al que pertenece antes de terminar.

La secuencia de Collatz de un número entero positivo n se obtiene usando la siguientes reglas:

- i. Si el número es par, se divide entre 2.
- ii. Si el número es impar, se multiplica por 3 y se suma 1.

La conjetura de Collatz dice que siempre alcanzaremos el 1 y a la cantidad de términos que componen la secuencia, desde el número inicial hasta el 1, se le llama la **longitud**.

Por ejemplo, la secuencia de Collatz del número 11 es 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1 y tiene una longitud de 15 términos.

4. Entregables: Elaborar un reporte (en formato pdf) en el que:

- i. se describa la solución a cada uno de los problemas,
- ii. se incluya el programa fuente de cada uno de los problemas.
- iii. se incluyan los enlaces a gdb online en donde se encuentren sus códigos.

5. Entrega: subir el pdf de su reporte en el enlace previsto para tal efecto en el aula virtual.

Elaboró: Elizabeth Pérez Cortés