Computación Concurrente 2017-2. Práctica 4: El problema de los filósofos.

José David Flores Peñaloza Susana Hahn Martín Lunas
Armando Ballinas Nangüelú

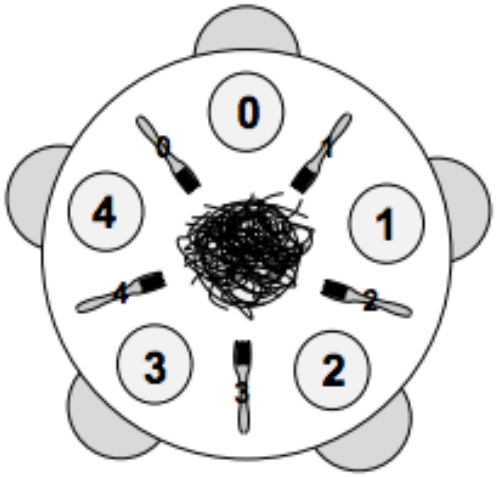
Fecha de entrega: 18 de abril de 2017.

1. El problema.

El problema de la cena de los filósofos fue propuesto por Edsger Dijkstra en 1965 y la descripción más común se basa en una mesa circular con cinco platos, cinco tenedores (o palillos) y un gran plato de espagueti al centro. Cinco filósofos, que representan la interacción de hilos, se encuentran sentados en la mesa y ejecutar el siguiente código:

```
1 while True:
2  think();
3  get_forks();
4  eat();
5  put_forks()
```

Los tenedores representan los recursos compartidos en donde cada hilo debe de conseguirse de manera exclusiva para poder realizar un progreso. Sin embargo cada filósofo necesita dos de ellos para comer, por lo que posiblemente deba esperar para poder obtener ambos. En la siguiente figura se encuentra una descripción gráfica de la mesa.



El problema se puede generar de modo que haya n filósofos.

1.1. Protocolo de los tenedores.

Para poder comer, cada filósofo debe tomar ambos tenedores y el protocolo es el siguiente:

```
1 Tomar tenedor derecho
2 Si el tenedor izquierdo está libre lo toma
3 Si no, espera hasta que se libere
```

Cuando un filósofo tiene ambos tenedores puede comer por un tiempo indefinido pero siempre terminará de comer. En cuanto termina de comer procede a soltarlos siguiendo el siguiente protocolo:

```
1 Suelta el tenedor izquierdo
2 Suelta el tenedor derecho
```

2. Ejercicios.

La práctica se puede realizar en parejas y se debe enviar al correo: *armandoballinas@ciencias.unam.mx* con el asunto *concurrente_p4*. Los archivos de la práctica deben estar dentro de una carpeta con el nombre de alguno de los integrantes del equipo y en el README deben venir los datos de ambos integrantes.

1. Hacer un programa utilizando Pthreads que simule el comportamiento de los filósofos. Deberán sincronizar el uso de los tenedores. El programa deberá recibir como argumento un número n que sea el número de filósofos en ejecución.

   Cada que un filósofo coma, se deberá imprimir que el filósofo está comiendo así como su número de identificación, por ejemplo: `El filósofo 3 está comiendo`. Para que la pantalla no se sature, deben asegurar que una vez que un filósofo come, espera al menos 2 segundos antes de intentar comer de nuevo.

   En esta implementación puede ocurrir un *deadlock* si los n filósofos intentar comer simultáneamente.

   Escribir este ejercicio en un archivo `filósofos.c`.

2. Mejorar la implementación de modo que se evite el *deadlock* sabiendo que si a lo más $n-1$ filósofos intentan comer simultáneamente, entonces no se producirá uno. En este caso si los n intentan comer al mismo tiempo, deben hacer que uno espere hasta que se liberen los tenedores necesarios y después debe intentar tomarlo de nuevo.

   Observen que lo que se debe garantizar es que a lo más $n-1$ filósofos comen en todo momento.

   Escribir este ejercicio en un archivo `filósofos_sin_deadlock.c`.

3. Argumenten, en el README, que si un filósofo es zurdo, es decir, que si existe uno cuyo protocolo para tomar los tenedores consiste en tomar primero el de la izquierda y luego el de la derecha, entonces no hay *deadlock*.

   Además, deben modificar el primer código para implementar esta modificación. Escribir este ejercicio en un archivo `filósofos_izq.c`.