Ejercicios propuestos

Tema 4 – Programación con threads

1. Escriba un programa en C llamado *orden* que cree dos threads y mediante el uso de un semáforo se fuerce que siempre uno de ellos escriba por pantalla antes que el otro.

2. Escriba un programa en C llamado *mutex* que cree dos threads que accedan simultáneamente a un buffer de 10000 enteros. Uno de ello lee en el buffer y el otro escribe en el mismo. El thread escritor, debe escribir el mismo valor en todos los elementos del buffer incrementando en uno el valor en cada pasada. El thread lector debe ir comprobando que todos los números del buffer son iguales, mostrando un mensaje de error en caso contrario o un mensaje de correcto si la condición se cumple. El código a realizar utilizará un mutex para acceder al bufffer si se indica un parámetro al ejecutar el programa. En caso contrario, los threads accederán al buffer de cualquier manera.

3. Escriba un programa en C concurrente que modelice la cena de los Filósofos. Se requiere que esté libre de interbloqueos (en las transparencias de ayuda se proponen varias alternativas para conseguir esto) y que haga lo siguiente:

   - El programa principal creará 5 semáforos que se utilizarán para modelar los recursos compartidos (los 5 palillos del problema de los filósofos)

   - El programa principal creará 5 hebras que ejecutarán una misma función. Después de crear las hebras, se meterá en un bucle infinito de espera (while(1);)

   - Cada una de las hebras es un filósofo que tiene que hacer lo siguiente:

     - Imprimir un mensaje por pantalla tipo “Filosofo i pensando”, siendo i el número de filósofo.

     - Esperar con un sleep un tiempo aleatorio entre 1 y 5 segundos.

     - Imprimir un mensaje por pantalla tipo “Filosofo i quiere comer”

     - Intentar coger los palillos que le corresponden. (El filósofo 0 necesita los palillos 0 y 1, el filósofo 1 los palillos 1 y 2, etc)

     - Cuando tenga el control de los palillos, imprimir un mensaje en pantalla tipo “Filosofo i comiendo”

     - Esperar con un sleep un tiempo aleatorio entre 1 y 5 segundos.

     - Soltar los palillos.

     - Volver al primer punto.

La salida del programa debería ser algo del tipo:

```
Filósofo 2 pensando
Filósofo 3 pensando
Filósofo 4 pensando
Filósofo 1 pensando
Filósofo 0 pensando
Filósofo 0 quiere comer
Filósofo 0 comiendo
Filósofo 2 quiere comer
Filósofo 2 comiendo
Filósofo 4 quiere comer
Filósofo 3 quiere comer
Filósofo 1 quiere comer
Filósofo 2 pensando
Filósofo 0 pensando
Filósofo 4 comiendo
Filósofo 1 comiendo
```

4. Escriba una solución al problema de los filósofos presentado anteriormente que garantice que no se produce interbloqueo utilizando mútex con condiciones (se pueden utilizar todas las que sean necesarias).

   De paso, se debe ampliar la información que se muestre por pantalla del siguiente modo: después de cada mensaje (pensando, comiendo, esperando) imprimir también el estado en el que se encuentran todos los filósofos. A continuación se muestra un ejemplo de ejecución:

```
Filósofo 0 pensando
Estado: 0 => pensando 1=> pensando      2=> pensando   3=> pensando   4=> pensando
Filósofo 1 pensando
Estado: 0 => pensando 1=> pensando      2=> pensando   3=> pensando   4=> pensando
Filósofo 3 pensando
Estado: 0 => pensando 1=> pensando      2=> pensando   3=> pensando   4=> pensando
Filósofo 4 pensando
Estado: 0 => pensando 1=> pensando      2=> pensando   3=> pensando   4=> pensando
Filósofo 2 pensando
Estado: 0 => pensando 1=> pensando      2=> pensando   3=> pensando   4=> pensando
Filósofo 0 quiere comer
Estado: 0 => esperando 1=> pensando     2=> pensando   3=> pensando   4=> pensando
Filósofo 0 comiendo
Estado: 0 => comiendo 1=> pensando      2=> pensando   3=> pensando   4=> pensando
Filósofo 3 quiere comer
Estado: 0 => comiendo 1=> pensando      2=> pensando   3=> esperando 4=> pensando
Filósofo 3 comiendo
Estado: 0 => comiendo 1=> pensando      2=> pensando   3=> comiendo   4=> pensando
Filósofo 4 quiere comer
Estado: 0 => comiendo 1=> pensando      2=> pensando   3=> comiendo   4=> esperando
Filósofo 0 pensando
Estado: 0 => pensando 1=> pensando      2=> pensando   3=> comiendo   4=> esperando
```