Práctica 4

Alejandro Guitarte Fernández

Noviembre 2022

1 Ejercicio 1

Para el intento 3 se produce un interbloqueo eventualmente. Esto es obvio, ya que si se da una iteración del bucle while(true) en la que justo los dos hilos pongan la variable *want* a true a la vez, ninguno de los dos podrá salir del bucle de espera ocupada, lo cual hace que se produzca el interbloqueo.

En el intento 4 también se producen interbloqueos. Si se da el entrelazado "perfecto", en el cual los dos cambios a false y true de las variables *want* se producen a la vez, o al menos antes de que el otrto hilo vuelva a la comprobación del bucle while, se volverá a entrar en el bucle, y así produciendo el interbloqueo.

2 Ejercicio 2

En este caso, obtenemos el resultado esperado, ya que sustituimos los cambios constantes en las flags *want* por un testigo similar al empleado en el intento 1. Sin embargo, no tenemos alternancia estricta, ya que también contamos con las flags mencionadas anteriormente. Hasta que paremos la ejecución del programa (pues estamos en un while(true)), se irá imprimiendo el nombre de hilo de uno y de otro, sin mezclarse las salidas entre ellos y sin producirse interbloqueos. Hemos comprobado experimentalmente que el algoritmo de Dekker es correcto.

3 Ejercicio 3

Realmente es muy parecido al algoritmo de Dekker, solo que ahora el testigo guarda el último hilo ejecutado, y agrupamos las condiciones para hacerlo más legible. Al igual que en el caso del algoritmo de Dekker, se obtienen los nombres de los hilos p y q sin intermezclarse entre ellos. Hemos comprobado también que el algoritmo de Peterson es, siendo más sencillo de implementar (menos bucles y condiciones anidadas), igualmente correcto.