Arthur Font Gouveia 20222613

1.a) Estas dos variables sirven para que múltiples hilos puedan coger turnos ordenados y que la condición de espera activa (línea 10) sea propia de cada hilo. La variable "turn" es global porque guarda el número de turnos asignados entre todos los hilos. La variable "turn_meu" es local porque guarda un número de turno en cada hilo.

b) La variable global "turn_actual" guarda el valor del turno del último hilo que ha entrado en la sección crítica (línea 11), para que así cuando se llame a la función "unlock" se indique el hilo que quiere salir de la sección crítica mediante la instrucción de la línea 15 y se dé permiso al siguiente hilo mediante la instrucción de la línea 16.

c) La línea 15 sirve para indicar que el hilo que estaba dentro de la sección crítica está saliendo. La línea 16 sirve para dar permiso al siguiente hilo. Con eso, y con la resta del código, se consigue que cada hilo que haya solicitado el "lock" entre de manera ordenada en la sección crítica (línea 11).

d) Sí, porque la operación "turn++" también es una sección crítica ya que múltiples hilos podrían acceder para escritura a la vez si la instrucción no fuera atómica. Un ejemplo sería en el caso de que suceda un cambio de contexto después de la ejecución de la primera instrucción de la línea 8, así el siguiente hilo puede coger el mismo turno si la variable "turn" aún no ha sido incrementada. Por lo tanto podrá haber más de un hilo dentro de la sección crítica y así un funcionamiento incorrecto del código.

e) El código funcionará para un número de hilos inferior a N pero no funcionará para un número de hilos superior a N. Esto se debe a que la estructura de control de los "flags" tienen reservado un vector de N posiciones. Por eso, en el caso de que más de N hilos cojan su turno, más de un hilo tendrá asignado el mismo valor de "turn_meu" (línea 9) y consecuentemente más de un hilo accederá a la sección crítica en el mismo tiempo (línea 11).

2.a) El "mutexD" sirve para señalizar si hay mujeres o no en el refugio. El "mutexH" sirve para señalizar si hay hombres o no en el refugio. La "dutxa" sirve para permitir o señalizar el acceso a la ducha o para hombres o para mujeres.

b) Sí, es necesario. Si "mutexD" o "mutexH" fueran un semáforo general, podría pasar el caso de que hombres y mujeres estarían se duchando juntos. Un ejemplo sencillo sería en el caso de que una mujer se esté duchando, llega un hombre y se espera en el semáforo de la ducha, pero si viene otro hombre este entraría en la ducha porque su semáforo no estaría bloqueado. Si "dutxa" fuera un semáforo general, una vez llegue un hombre y una mujer, ambos accederían a ducha a la vez.

Arthur Font Gouveia 20222613

2. c) Una mujer se esperará al semáforo de la ducha (línea 8) y las otras dos se esperarán en el semáforo de las mujeres (línea 6). Porque cuando una mujer entra en la sección crítica, se actualiza el número de mujeres a uno y como se cumple la condición de la línea 8, ella se quedará esperando allí. Las demás no entrarán en la sección crítica porque la mujer que ha entrado aún no ha salido, por lo tanto esperarán en la línea 6.

d) Depende. Si el hombre que se está duchando aún no ha acabado (ejecutado la línea 22), el nuevo hombre podrá entrar a la ducha. Pero si no hay más hombres se duchando, o sea, si el hombre se duchó (línea 22) antes de que el nuevo hombre llegó, el nuevo hombre tendrá que esperar hasta que no haya ninguna mujer.

e) ~~Se debería adicionar la instrucción "sched(homes<4);" entre la línea 8 y 9 y adicionar la instrucción "sched(homes)~~

~~Se debería adicionar la instrucción~~

Se debería reemplazar la línea 9 por la instrucción "if(dones<4) sem_post(&mutexD)" y reemplazar la línea 20 por la instrucción "if(homes<4) sem_post(&mutexH)". De esta forma se bloquea la entrada simultánea de más de cuatro mujeres en la línea 10 y también se bloquea la entrada simultánea de más de cuatro hombres en la línea 21.

Arthur Font Garcia 20222613

3. a) Sí, es necesario proteger las dos partes porque ambas acceden a variables globales para escritura. No se podría proteger las dos partes con llaves diferentes por ~~el~~ mismo motivo ya que ambas partes acceden para escritura en las variables globales "client" y "ocupat".

b) No hace falta el while porque no es necesario volver a comprobar la condición de que no hay clientes (línea 12), una vez que todos los clientes que llamaron la función "client" cortaron el pelo y solo hay un barbero.

c) No, porque solo hay un barbero y el único barbero atiende solamente un cliente a la vez, así si el ha notificado que está libre a un cliente específico de la cola, no hace falta volver a comprobar esta condición para seguir con el código.

d) La llamada de la línea 14 se hace después que el barbero recibe la notificación de que hay cliente. La llamada de la línea 25 se hace después que el cliente recibe la notificación de que el barbero está libre. La línea 14 tiene el propósito de despertar el primer cliente de la cola y la línea 25 de despertar el barbero.
Sí, es necesario que el barbero esté esperando al "wait" porque si no el primer cliente irá ejecutar la función de la línea 29 y no habrá barbero para cortar su pelo.

e) Se debería cambiar el "if" de la línea 12 por un "while", y se debería cambiar la variable "ocupat" por un vector en que cada posición corresponda a la estado de cada barbero, haciendo las modificaciones necesarias en el código.