Documentación Caso 1

Nicolás Chaves de Plaza - 201313618
Sebastian Sierra Cuervo - 201125266

Funcionamiento Global

Descripción paso a paso de la ejecución del programa:

-El inicio de la ejecución del programa, tiene lugar en la clase Principal del mismo, en esta se hace la lectura del archivo de propiedades y además se inicializan y empiezan todos los threads con los valores previamente recibidos. Estos threads corresponden tanto a los clientes como a los servidores de la aplicacion.
Adicionalmente, en esta clase se inicializa el buffer que será el lugar de interacción de ambos actores.
-En cuanto al buffer como tal, este es el lugar donde se relacionan tanto productores(Clientes) como consumidores(Servidores). En este hay métodos que facilitan el depositar un mensaje así como sacarlo y modificarlo.
-Una vez que todos los threads han sido inicializados, el programa entra a funcionar. Los clientes depositan sus mensajes en el buffer de manera concurrente al tiempo que los productores los sacan y los modifican.
-Una vez todos los mensajes han sido atendidos, se despierta a todos los servidores (para que no quede ninguno dormido que no termine ejecución).

Nota: Para verificar el correcto funcionamiento del programa, en todo momento se imprime en consola el cliente junto con su respectivo mensaje cuando entra y el mismo cliente cuando sale con el mensaje modificado. Todos los clientes que entraron deben salir y todos los mensajes deben ser procesados por un servidor.

Interacciones y elementos relevantes

Clases

Principal

Esta clase se encarga de leer del archivo de propiedades el número de clientes, servidores, mensajes y el tamaño del buffer. Adicionalmente se encarga de la inicialización de los threads así como del inicio de la ejecución del programa.

Buffer

Es el lugar donde interactúan los clientes con los consumidores. En este, los primeros depositan los mensajes para luego ser retirados y procesados por los segundos.

Cliente

Son los encargados de entregar los mensajes al buffer y esperar por la respuesta del servidor. La clase extiende de thread ya que van a haber múltiples hilos de ejecución concurrentes.

El run de la clase se encarga de crear el mensaje nuevo, depositarlo en el buffer y luego de que el proceso ha culminado para el número de mensajes por cliente, disminuir el número de clientes en 1.

Servidor

Estos son los encargados de atender los mensajes de los clientes. La clase extiende de threads ya que se van a tener múltiples hilos de ejecución simultáneamente.

En el método run lo único que se hace es activar el método retirar del buffer mientras el número de clientes presentes sea mayor a 0.

Mensaje

Es la clase donde se almacena el mensaje que en este caso consiste en un número. Adicionalmente tiene el método usado por los servidores para incrementar dicho número en 1.

Interacciones Relevantes

Cliente-Buffer

El cliente se relaciona con el buffer a través del método depositar. Este se encarga de poner el mensaje de la cola durmiendo al cliente hasta que el anterior sea procesado.

En este método se ven elementos importantes:

- Al inicio el método Yield que cede el procesador en cada iteración, esta es la espera activa de ser atendidos de los clientes.
- Un bloque sincronizado que se establece como un candado dentro del cual se agrega cada mensaje a la cola y se disminuye su capacidad. Además, se notifica a todos los servidores para que empiecen a buscar mensajes que atender.
- Finalmente, se tiene otro candado para ciertos casos cuando el servidor no alcanza a quedarse dormido. Este en un bloque sincronizado sobre mensaje.

Servidor-Buffer

El servidor se relaciona con el buffer a traves del metodo retirar. Este se encarga de retirar de la cola los mensajes para poder ser atendidos y devueltos al cliente correspondiente. Este método es sincronizado por lo que se usa el concepto de exclusión mutua para el manejo de la concurrencia.

En este método se ven elementos importantes:

- Se tiene un bloque que revisa si hay mensajes para poder proceder a retirarlo, si no, se espera a que llegue alguno usando el método wait.
- Cuando se atiende el mensaje, se cambia su valor y se procede a notificar al cliente correspondiente para seguir con otros mensajes.

Servidor-Mensaje

Cada cliente tiene un mensaje. Adicionalmente, dentro de este último, se encuentra el método sincronizado(exclusión mutua) que incrementa su valor.

Servidor-Cliente

El servidor y el cliente se relacionan como se habia visto anteriormente a través del buffer que deposita los mensajes de uno y le permite al otro atenderlos.