

# **SISTEMAS OPERATIVOS**

## **PROYECTO PRIMER PARCIAL**

### **Integrantes:**

- Adrián Aguilar
- Cristina Barreno
- Marlon Loayza

### **Introducción**

El manejo de hilos es algo común en los sistemas operativos actuales, para lo cual un manejo apropiado es imprescindible, para realizar esto se han diseñado e implementado varios mecanismos de sincronización, los cuáles han sido estudiados durante la clase de Sistemas Operativos.

Se ha diseñado un programa Lector de RSS en Java, en el cuál se ha utilizado el paradigma productor/consumidor, en donde tenemos para cada Proveedor de RSS, un Hilo Productor encargándose de él y un Hilo Consumidor. Asignándoles así a cada Proveedor que llega su hilo productor respectivo.

Para la implementación del Lector hemos utilizado una Cola que es compartida tanto para los hilos productores como para el hilo consumidor, por lo que se observa este recurso compartido está siendo accedido por los diferentes hilos de la aplicación, al tener esta situación observamos que se presentan condiciones de carrera, por lo que se ve en la necesidad de implementar algún mecanismo de sincronización de procesos.

### **Metodología**

Para la implementación del proyecto, se trabajó con una Cola para guardar todos los post disponibles en los proveedores de RSS, se utiliza una Cola para poder mostrar los Posts más recientes al inicio de la aplicación, y los más antiguos al final.

Como esta Cola está siendo compartido con varios hilos se utilizó semáforos para sincronizar el acceso a la Cola.

Se han asignado un máximo de 10 hilos que se encargaran de los proveedores de RSS para recoger la información necesaria a ser ingresada en la Cola. Es decir en total se podrán manejar hasta un máximo de 10 Proveedores de RSS.

Los URL de los Proveedores se guardan en una LinkedList, debido a que debemos mantenerlos almacenados para cuando necesitemos hacer un refresh en nuestra aplicación y ver los nuevos posts que se hayan agregado.

Los métodos que han sido sincronizados, se encuentran en la clase ProductorConsumidor para obtener información y añadir información a la Cola son:

**public synchronized Queue getData() - public synchronized void addQueue(String url)**

El proyecto consta de 11 clases, entre las que destacan las clases ProductorConsumidor, Productor, Consumidor, XMLParserRSS.

Se desarrolló una interfaz gráfica sencilla para usar la aplicación, en donde constan 2 botones, el botón **Agregar** nos permitirá añadir a la lista los URL de los Proveedores de RSS que se coloquen en el TextField y ver los posts publicados en la aplicación, y el botón **Refrescar** nos permitirá actualizar para observar los últimos cambios de los proveedores de RSS. Así mismo se muestra un cronometro que indica el tiempo que se tomara en hacer el Refresh automático.

También en el menú tenemos la opción para Configuración, con la cuál podremos programar el tiempo para hacer el refresh y la opción acerca de para ver los miembros del grupo.

### Resultados

- Se logró implementar los mecanismos de sincronización de procesos con éxito para la utilización del Recurso Cola o Queue.
- La implementación de la aplicación soporta un máximo de 10 hilos correspondientes a 10 productores RSS.
- Se usó la Estructura de Datos Queue o Cola, para poder mostrar los posts más recientes al inicio.
- Se puede hacer una actualización o Refresh de forma automática o manual. Además el usuario podrá configurar el tiempo para hacer el Refresh automático.

### Conclusiones

- Se aprendió la implementación de mecanismo para sincronización de hilos en Java.
- Se nos hizo complicado el manejo de hilos, antes de haber usado los mecanismos de procesos.
- El proyecto no es tan flexible puesto que no se alcanzó a implementar un pool de hilos, para soportar n hilos, es por esto que solo se puede usar un máximo de 10 hilos productores en la aplicación.

### Referencias

1. [http://www.chuidiang.com/java/hilos/wait\\_y\\_notify.php](http://www.chuidiang.com/java/hilos/wait_y_notify.php)
2. <http://jdeveloper.wikispaces.com/14.2.-+Programaci%C3%B3n+Multihilo>
3. <http://javabasico.osmosislatina.com/curso/progfinal2/threads2.htm>
4. <http://www.vogella.com/tutorials/RSSFeed/article.html>