**ESCUELA SUPERIOR POLITECNICA DEL LITORAL**

**Proyecto Sistemas Operativos**

**Título del Proyecto: RSS reader multi-hilos.**

**Descripción y alcance del proyecto:**

Se debe diseñar e implementar un RSS reader que permita ver los feeds a los que se  subscribe el usuario. El cliente debe ser multi­hilos, con un hilo para cada proveedor RSS.  Ese hilo se encarga de contactar al proveedor RSS y coloca en un buffer las entradas/posts  que se descarga, usando el paradigma productor­consumidor. Estos son hilos productores.    Otro hilo toma los mensajes de este buffer y los muestra al usuario usando una interfaz  amigable (colores, etc.). Este es un hilo consumidor.    Los mensajes se deben mostrar en pantalla conforme van llegando, pero deben implementar  un archivo de configuración en el que se puedan definir parámetros como: qué tan seguido  se conecta al RSS feed un hilo productor. Para los casos en que el usuario seleccione un  tiempo grande entre “pulls” (por ejemplo, 10 minutos), el cliente debe tener un mecanismo (por ejemplo, vía botón o combinación de teclas especiales) mediante el cual se puede  “refrescar” el feed (es decir, forzar un pull de los posts de los feeds a los que está suscrito el  usuario).

**IMPORTANTE:** Su diseño e implementación NO debe ser susceptible a condiciones de  carrera. Por esta razón, tengan cuidado de seleccionar y utilizar los mecanismos adecuados  de sincronización para proteger los datos compartidos entre los hilos.

**Metodología:**

Para la implementación de este proyecto se utilizó 3 clases, para tomar el archivo xml desde el servidor RSS, parsear el xml extrayendo el texto de la noticia y generar la lista de noticias.

El control de sincronización de los hilos esta esquematizada con la metodología productor, consumidor y un buffer, que evitara las condiciones de carreras al entrar en la sección critica de pull y push del buffer que contendrá la lista de noticias que se actualiza cada periodo de tiempo.

**Paradigma productor-consumidor:**

Se iniciara un nuevo hilo productor cada que el usuario se suscriba a un canal, el usuario no puede ingresar un mismo canal RSS dos veces esta validado para rechazar la suscripción en caso de que lo haga, los canales a los cuales se suscribe el cliente aparecerán en una lista de suscripciones al lado izquierdo de la pantalla.

El **hilo productor** se ejecutara cada cierto tiempo definido por default en el código, una vez que se inicie el hilo se conectara al servidor RSS descargando el archivo, para proceder a la lectura del xml extracción de las noticias con título, descripción, link y autor, que están contenidas en los tags del xml. Una vez extraída la noticia se la agregara a una lista de String.

Luego de tener la lista de noticias del hilo con su RSS correspondiente se procede a escribir en el buffer, realizando push en este momento empieza la sección critica se bloquea el ingreso de otro hilo dándole valor false a la variable disponible del buffer, en caso de que llegue otra petición de push o pull mientras se ejecuta la sección critica deberá esperar a que termine de ejecutarse el hilo que está haciendo push en ese momento, una vez dentro de la sección critica del buffer se valida que las noticias que se van ingresar no existan en el buffer, con esto evitamos noticias duplicadas para posteriormente escribir en el buffer las noticias nuevas, y colocar la variable disponible como true, en este momento liberamos el recurso para que otro hilo ingrese y efectúe la operación correspondiente sea push o pull.

El **hilo consumidor** se ejecutara cada periodo del tiempo desde que se inicia el programa, este periodo de tiempo también está definido por default dentro del código, la función del hilo consumidor será hacer pull al valor del buffer cada cierto tiempo y actualizar el textarea de nuestra interfaz, en el momento que se llama al pull se bloquea la variable disponible dentro del buffer para que no pueda efectuarse push mientras se consume el recurso, los hilos que ingresen mientras se realice un pull entraran a estado wait hasta terminar el pull, una vez terminada la operación, estará disponible nuevamente el acceso al buffer para realizar los push.

Además de ejecutarse cada cierto tiempo para actualizar las noticias, existe la opción de actualizar mediante el botón en la interfaz, esta acción detendrá el hilo consumidor que se ejecuta al iniciar el programa y creara una nueva instancia que inmediatamente consumirá el recurso y lo mostrara por pantalla.

**El buffer** como se mencionó anteriormente controlará el acceso a las secciones críticas tanto de los hilos productores y consumidores mediante la variable disponible, que bloquea o libera el buffer si está o no siendo utilizado.

**Resultados:**

Se obtuvo como resultado una aplicación de lector RSS, que consume cada cierto tiempo archivos desde el servidor RSS al cual fue suscrito, se puede suscribir a varios RSS, cada suscripción es un hilo independiente que se corre de fondo y se actualiza la información mostrándola por pantalla. Fue implementado con el paradigma productor – consumidor para evitar condiciones de carrera al momento de sincronizar la información, tanto para guardar o tomar las noticias guardadas en un buffer.

**Conclusiones:**

El uso de un método de sincronización, resulta útil en el manejo de procesos multihilos. A lo largo del desarrollo de esta aplicación se notó gran diferencia al ejecutar el lector RSS sin el paradigma productor/ consumidor saltaban muchos errores, al momento de escribir en el buffer, o de tomar la información del mismo.

Una vez implementado las condiciones dentro del buffer al momento de consumir o colocar información, se evitó este tipo de errores y las condiciones de carrera. Aunque el tiempo de ejecución de los hilos se tardaban un poco al intentar modificar o tomar la información del buffer si este estaba haciendo ocupado por otro hilo, pero a cambio se obtuvo un correcto funcionamiento y ejecución de todos los hilos así tengan que esperar hasta que se libere el recurso, resultando favorable la implementación del paradigma.