Tercera parte del proyecto programado

Juan Carlos Porras Quirós[1], Dagoberto Quirós Vargas[2], Pablo Sauma Chacón[3]
CI-1323 Arquitectura de computadores
Escuela de Ciencias de la Computación e Informática
Facultad de Ingeniería
Universidad de Costa Rica
[1]*b25201@ecci.ucr.ac.cr*, [2]*b05051@ecci.ucr.ac.cr*, [3]*b16195@ecci.ucr.ac.cr*

Noviembre de 2015

Capítulo 1

Manual de Instalación

Para poder ejecutar el programa se requiere descargar el archivo y descomprimirlo donde se desee. Una vez hecho lo anterior tenemos tres opciones para ejecutar el programa:

1.1. Sin compilar

Al archivo llamado *Main* hay que cambiarle el nombre por *Main.exe*, al archivo *Arquicomp* por *Arquicomp.jar* y los archivo *AutoTestRapido* y *AutoTestLento* por *AutoTestRapido.bat* y *AutoTestLento.bat* respectivamente; esto pues los archivos compilados y de tipo *batch* no se pueden enviar por correo electrónico por motivos de seguridad.

1.2. Compilando

También, se puede proceder compilando los archivos en la computadora en que se descarguen. Para esto, se requiere el compilador de C++ de GNU y la biblioteca *Pthreads*, se recomienda utilizar la siguiente línea para que los archivos *batch* funcionen correctamente:

```
1  g++ -fpermissive -pthread Main.cpp -o Main
```

1.2.1. Con archivos batch

Ahora bien, se puede ejecutar el programa mediante los archivos de tipo *batch*, pero estos sólo sirven para ejecutar el programa en consola. El *batch AutoTestRapido.bat* para ejecutar el archivo en modo rápido, o el archivo *AutoTestLento.bat* para ejecutar en modo lento. El modo lento también se puede ejecutar mediante el siguiente código:

```
1  Main.exe tr 30 1 0 Hilos/1.txt Hilos/2.txt ...
```

Y el modo rápido:

```
1  Main.exe fr 30 1 0 Hilos/1.txt Hilos/2.txt ...
```

Para cada uno de los casos anteriores *Main.exe* es el ejecutable de C++, t quiere decir que se ejecutará el programa con el modo lento y f que se ejecutará en modo rápido, r que despliegue texto, luego los tres números que vienen son el *quantum*, m y b, por último se agregan los archivos de los "hilillos"; nótese que para cada "hilillo"se especifica el directorio en el que está contenido y luego el nombre del archivo.

1.2.2. Directamente mediante consola

Si se desea trabajar directamente mediante consola, se puede insertar los comandos igual como se especifican en *1.2.1 con archivos batch*, donde se especifican todos los parámetros desde el inicio, o bien se puede llamar simplemente a *Main.exe* y brindarle los datos conforme el programa los vaya solicitando.

1.3. Interfaz gráfica

Ya sea que se compiló o se utilizó el archivo que traía el comprimido, se puede ejecutar el archivo *Arquicomp.jar*, el cual es una interfaz gráfica hecha en Java, la cual se encarga de solicitar los parámetros y archivos para con estos llamar al ejecutable de C++. Se ve así:

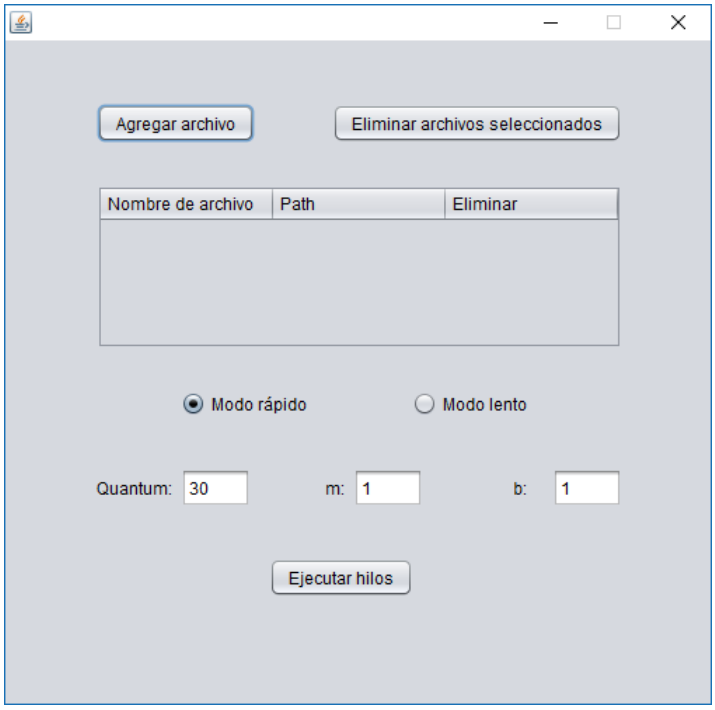

Figura 1.1: Pantalla de inicio

Para utilizar ésta interfaz se presiona el botón *Agregar archivo* para navegar por el computador y seleccionar uno o más "hilillos"para ejecutar. Una vez seleccionados, los archivos aparecen en la tabla, como lo muestra la *Figura 1.2*. Si se desea eliminar uno o varios de los "hilillos", se pueden seleccionar en la tabla y se presiona el botón *Eliminar archivos seleccionados*, como también lo muestra la *Figura 1.2*.

Por defecto, el sistema tiene los parámetros *Quantum*, m y b en 30, 1 y 1, respectivamente; pero se pueden cambiar a necesidad. Por último, con los archivos que se quieren ejecutar seleccionados, se presiona el botón *Ejecutar hilos* y luego de un tiempo se desplegará una venta con el resultado de la ejecución de los "hilillos".

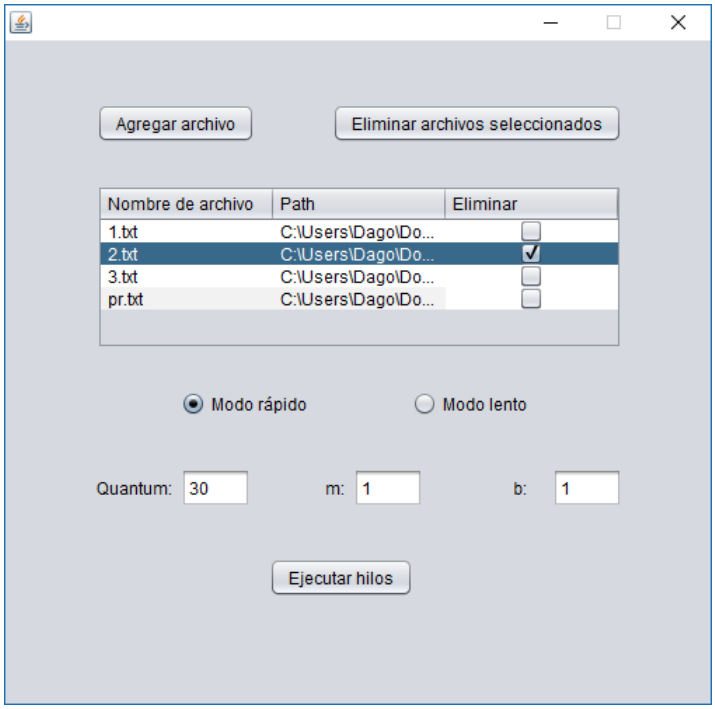

Figura 1.2: Pantalla de inicio

Capítulo 2

Diseño

Tras haber hecho una corrección de los fundamentos de diseño, se redacta un pseudocódigo que busca mostrar el flujo lógico de las instrucciones. Se presenta el pseudocódigo como explicación del paso a paso que implica cada una de las instrucciones implementadas:

2.1. LW

```
si el bloque esta C
       Implica que no ha sido modificado.
       Puede que alguien mas lo tenga (sin haberlo modificado), o puede que solo yo lo tenga
       Solo se lee directo de memoria y listo
       Es necesario bloquear la cache propia para que nadie ms la agarre, y se desbloquea a final
           de ciclo

sino, si el bloque esta en M
       Solo yo lo tengo. Puede que alguien mas lo tenga, pero esta invalidado, y eso corresponde a
           un miss.
       Solo se lee y listo

sino, es porque esta en I, o no lo tengo del todo (da igual)
       Se tiene que pedir el bus
       (Si no se logra obtener el bus, desbloqueo mi cache y termino el ciclo)
       Primero, hay que revisar si lo que esta en el bloque donde debera estar el bloque, esta M
       si esta modificado, hay que mandarlo a escribir (bloqueo mi cache, uso el bus para ir a
           memoria a escribirlo, y vuelvo)
       es necesario preguntarle a la otra cache a ver como lo tiene
       Se bloquea el cache ajeno

       si lo tiene modificado
              se manda a escribir a memoria el bloque (se hace similtaneo)
              luego se copia de la cache ajena a la propia
              Luego se libera el cache del otro
              Me devuelvo y libero el bus

       sino
              Se libera el cache del otro
              Se usa el bus para traer el bloque de memoria
              Se libera el bus
```

2.2. SW

```
si el bloque esta C
       Es necesario ir a invalidar el bloque del otro cache
       Pido el bus (Si no se logra obtener el bus, se queda esperando)
       Bloqueo la otra cache
       Cambio el bloque del otro de C a I
       Desbloqueo la otra cache
       Me devuelvo y libero el bus
       Cambio el bloque en mi cache a M
sino, si esta en M
       Solo yo tengo (valido; puede que alguien mas lo tenga, pero estaria invalidado, lo cual es
           un miss)
       Simplemente escribo
sino, es porque esta en I, o no lo tengo del todo (da igual, es un miss)
       Es necesario ir a revisar el bloque del otro cache
       Pido el bus (Si esta ocupado, lo libero)
       Bloqueo la otra cache
       Reviso el bloque
       si esta C
              lo pongo en I
              libero la cache
              me devuelvo
              libero el bus
              modifico el bloque y lo pongo en M
       si esta en M
              primero lo tengo que ir a escribir a memoria
              luego, ya cuando volvi, lo dejo en invalidado (previendo que lo voy a escribir yo)
              me devuelvo a mi propia cache
              modifico el bloque
       si esta en I
              Nadie lo tiene, asi que hay que ir a traerlo de memoria
              Ya cuando lo traje a mi propia cache, lo modifico
```

2.3. LL

Exactamente igual a LW, solo que después de que se ejecuta, pone en RL el valor de memoria.

2.4. SC

Si la direccion de memoria es igual al valor de RC, se ejecuta un SW. Si no, se pone un cero en RL.

Capítulo 3

Problemas no resueltos

Dada la naturaleza del proyecto, donde se tiene una interfaz gráfica en Java, pero la lógica y el procesamiento del sistema se encuentran en C++; no fue posible mantener una comunicación abierta constante entre ambas plataformas, es decir, no fue posible que las impresiones de *ciclo por ciclo* generadas cuando se ejecuta el sistema en modo lento, se muestren en tiempo real en la interfaz Java.

La implementación que se presenta busca arreglar esto, "simulando" el modo lento: al correr el modo lento, lo que el sistema hace, es ejecutar los hilos ciclo por ciclo, armando un búfer de todas las impresiones que se hacen en modo lento. Es decir, por dentro (sin ser visible al usuario) se genera la impresión del ciclo, acumulándose en un búfer y se ejecuta la siguiente iteración. Cuando se termina el programa, el búfer tiene las impresiones de todos los ciclos.

Para simular el modo lento, la interfaz Java procesa el archivo de texto, las divide y lo va mostrando ciclo por ciclo en la interfaz. Esto puede entenderse como "ver la repetición" de lo que se ejecutó en el sistema. El proceso en sí se hace en modo lento, pero lo que se le presenta al usuario es una transcripción de las impresiones del *ciclo por ciclo* obtenidas tras haber hecho todos los ciclos en modo lento.

Capítulo 4

Notas de los integrantes del equipo

Juan Carlos Porras Quirós	100
Dagoberto Quirós Vargas	100
Pablo Sauma Chacón	100