

Sistemas Operativos (SIU4085)

Proyecto 1- Febrero 2023

15%

Modificación Concurrente de una Imagen

1. *Objetivos del Proyecto*

- Resolver un problema concurrente/paralelo utilizando hilos.
- Estudiar el desempeño de la solución propuesta variando algunos parámetros de entrada.
- Comparar el desempeño de las soluciones secuenciales con la solución concurrente.
- Utilizar de forma adecuada llamadas al sistema relacionadas con el sistema de archivos.
- Comprender un formato simple de almacenamiento de imágenes

2. *Descripción General*

La idea del proyecto es modificar una imagen en formato **bmp**, aplicando filtros sencillos, en forma concurrente. Posteriormente se comparará el desempeño de las dos soluciones secuenciales con la solución concurrente desarrollada.

Cada grupo deberá desarrollar un programa concurrente que siga las siguientes especificaciones:

2.1 Parámetros de entrada

El programa concurrente se invocará de la siguiente forma:

\$./imgconc -i imagenIn -t imagenOut -o opción -h nhilos

Donde:

imgconc es el nombre del archivo ejecutable que el programa concurrente.

imagenIn: es el nombre del archivo de entrada. Es una imagen en formato **bmp**.

imagenOut: es el nombre del archivo donde quedará la imagen de salida (*note que en los programas que les doy como ejemplo, el nombre del archivo de salida es único y está codificado en el programa. Ud. debe modificar estos programas para que el usuario pueda ingresar un nombre cualquiera para el archivo de salida*)

opción: es un número entero relacionado con el tratamiento que se le aplicará a la imagen, así si la opción es:

1. Se aplicará la siguiente transformación a los bytes que conforman el pixel:

A cada byte del pixel se le transforma en:

$$\text{Pixel}[i] = ((R * 0.3) + (G * 0.59) + (B * 0.11)); \text{ // para } i \text{ entre } 0 \text{ y } 2$$

Donde *R* es el byte Red, *G* es el byte Green y *B* es el byte Blue

Esta es la transformación que viene en el archivo original BMP.c

2. Se aplicará el promedio: $\text{Pixel}[i] = (B + G + R) / 3$; // para *i* entre 0 y 2
3. Para la opción 3 los estudiantes buscarán un filtro sencillo para aplicar a la imagen.

Con la opción 1, la imagen de salida para la imagen de entrada, que se muestra en la figura 1 (a), la pueden observar en la figura 1(b).

n hilos: es el número de hilos concurrentes que procesará la imagen (para este programa el número siempre es mayor que 0. Si el valor del parámetro es 1, deben crear 1 hilo para realizar toda la tarea). Aunque hacer un programa con un hilo es equivalente a hacer un programa secuencial, siempre que hagamos referencia a programas secuenciales son aquellos que no tienen hilos.

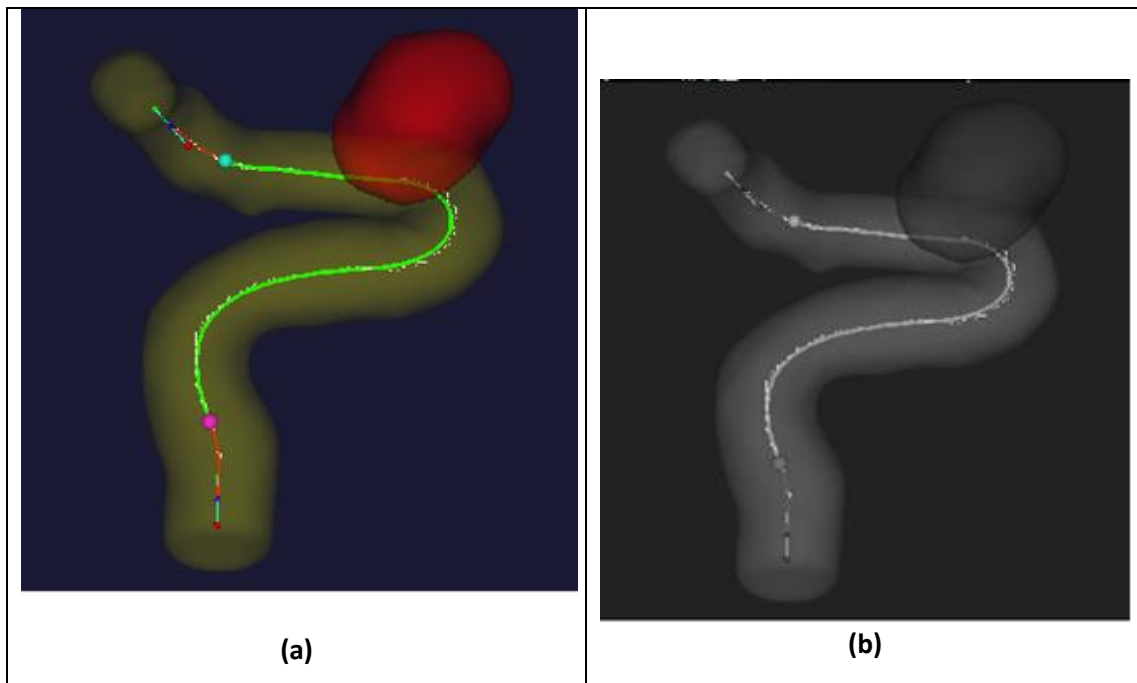


Figura 1: Imagen antes (a) y después (b) de realizar la transformación (Imagen tomada del proyecto Thrombus VPH del Ing. Leonardo Florez)

2.2 Formato de la imagen y archivos de apoyo

En formato **bmp**, las imágenes se guardan en archivos de texto. Tienen un *header* inicial con información de la imagen: tamaño, alto ancho, tipo de compresión, etc. En la Figura 2b, se puede observar el **header** representado con un rectángulo azul; este header está al principio de la imagen. A continuación, se almacena el conjunto de pixeles de la imagen. Cada pixel está representado por tres bytes (Figura 2a).

El programa que se encuentra en el archivo **BMP.c** (bajado de internet) lee una imagen en formato **bmp** y a cada pixel lo transforma en un solo valor, a medida que se va leyendo. La transformación se muestra a continuación.

```
fread(&B,sizeof(char),1, archivo); //Byte Blue del pixel
fread(&G,sizeof(char),1, archivo); //Byte Green del pixel
fread(&R,sizeof(char),1, archivo); //Byte Red del pixel
```

//Conversión a escala de grises

```
imagen->pixel[i][j]=(unsigned char)((R*0.3)+(G*0.59)+(B*0.11));
```

El programa BMP2.c es una modificación del anterior, realizada por la prof. M. Curiel, donde la imagen se lee completamente en una matriz de NxMx3 donde N es el alto de la imagen, M es el ancho y 3 es la cantidad de bytes por pixels. Una vez que la imagen está completa en la memoria, se llama a la función *convertir_imagen* para realizar la transformación mencionada anteriormente.

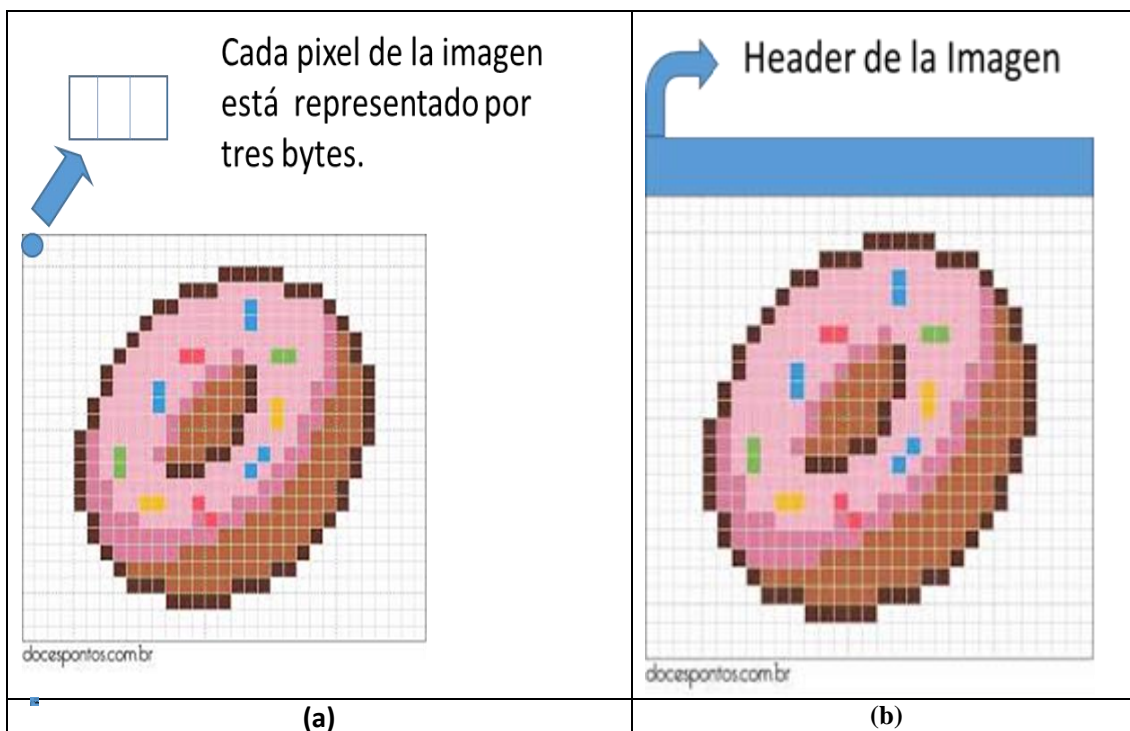


Figura 2: Composición de un pixel (a) y Header de la imagen (b)

2.3 Procesamiento concurrente de la imagen

De acuerdo con el número **nhilos** que se reciba como parámetro de entrada, se dividirá la imagen en partes iguales y se le dará un trozo a cada uno de los hilos para que apliquen el filtro o transformación seleccionado por el usuario. Una vez que todos los hilos terminan, la imagen se escribe en el archivo de salida (argumento que introdujo el usuario: **ImagenOut**). La forma como dividirán la imagen la seleccionan Uds., en la figura 3 a y b se muestran 2 ejemplos (los hilos son las líneas en azul). La idea, en lo posible, es dividir la imagen en trozos de tamaño similar.

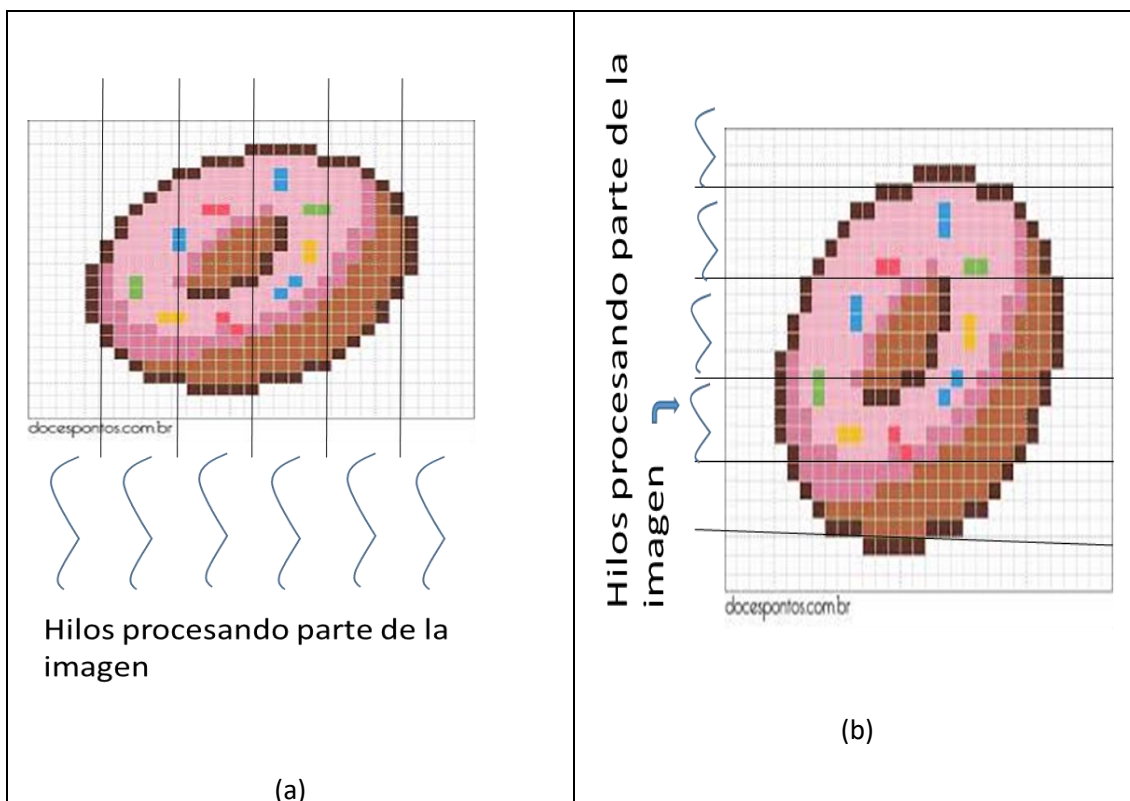


Figura 3: División de la Imagen entre los hilos.

Para el desarrollo del programa concurrente pueden utilizar los archivos **BMP.c** y **BMP2.c**. Modifíquenlos para adaptarlos a los nuevos requerimientos y dejen el nombre del (los) autor (es) y sus comentarios. El resultado del programa multihilos debe ser similar a la salida de BMP2.c

2.4 Formato del Archivo

Tanto el formato del archivo que contiene la imagen, como el código del archivo BMP.c y BMP2.c se les explicará en clase.

3. Pruebas de Rendimiento

Para la entrega los estudiantes realizarán las siguientes pruebas de rendimiento:

Experimento 1: Se comparará el desempeño de las dos soluciones secuenciales (contenidas en BMP.c y BMP2.c) versus la solución concurrente multihilos. Para ello:

- Revise los códigos y de acuerdo con los conocimientos que ha adquirido en la asignatura (y en otras asignaturas de la carrera), plantee al menos una o dos hipótesis sobre cómo será el comportamiento de los tres tipos de programas
- Las métricas para evaluar el desempeño de las aplicaciones serán: los tiempos de respuesta de los programas y el uso de la memoria.
- Las dos soluciones secuenciales las comparará únicamente con una solución concurrente que use 4 hilos.
- Use para el experimento la imagen de mayor tamaño sugerida en la tabla I.
- Los programas a utilizar para obtener las métricas de desempeño son: **time (tiempo total de ejecución)**, y el ejecutable generado a partir del programa **memory.c** para medir el uso de la memoria.
- Realice la medición y grafique los datos obtenidos. En la figura 4 pueden ver un ejemplo de gráfico. Los datos del gráfico son ficticios pero noten títulos en los ejes, unidades, etc.
- Analice los resultados obtenidos a la luz de las hipótesis planteadas y de los conceptos aprendidos en clase.

Nota: Para tomar el tiempo de ejecución de un comando puede usar el comando time de la siguiente forma:

\$ time imgconc imagenentrada.bmp imagensalida.bmp 2 4

Experimento 2: Estudie el desempeño de la solución concurrente variando el tamaño de la imagen y el número de hilos, tal y como se muestra en la Tabla 1. Planteen también una hipótesis de trabajo; realicen las mediciones indicadas en la tabla; grafiquen los resultados (similar a como se muestra en la figura 4) y analicen los resultados obtenidos.

	Tamaño de la imagen			
Numero de Hilos	Entre 80 y 100K	Entre 5 y 10MB	Entre 20MB de 40MB	Mas de 40MB
2	Tiempo total de ejecución, uso de Memoria	Tiempo total de ejecución, uso de Memoria	Tiempo total de ejecución, uso de Memoria	Tiempo total de ejecución, uso de Memoria
4	Tiempo total de ejecución, uso de Memoria	Tiempo total de ejecución, uso de Memoria	Tiempo total de ejecución, uso de Memoria	Tiempo total de ejecución, uso de Memoria
6	Tiempo total de ejecución, uso de Memoria	Tiempo total de ejecución, uso de Memoria	Tiempo total de ejecución, uso de Memoria	Tiempo total de ejecución, uso de Memoria

				Memoria
--	--	--	--	---------

Tabla I: Medidas a realizar sólo con el programa concurrente.

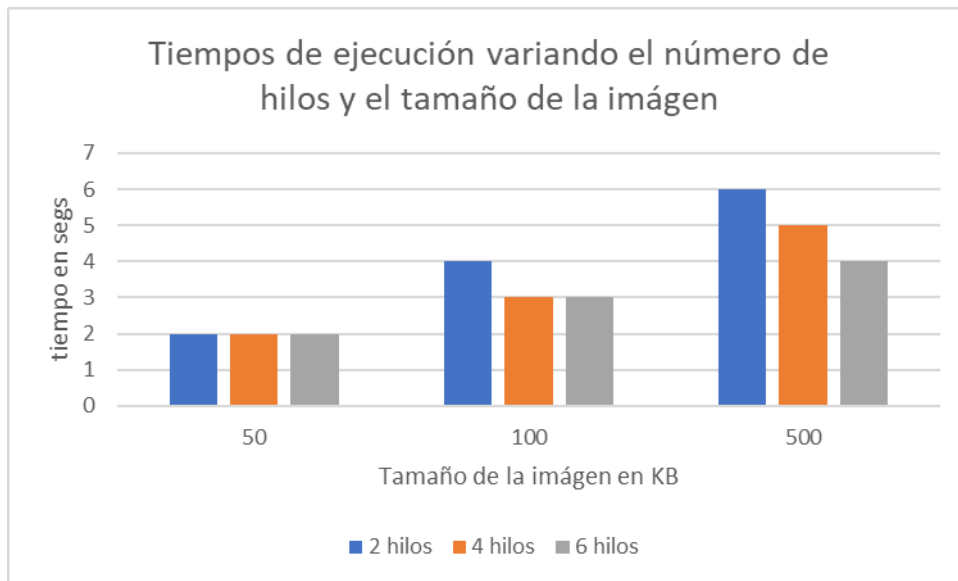


Figura 4: Ejemplo de Gráfico para comparar tiempo de ejecución de sol. Concurrente con varios parámetros.

4. Informe

Una vez realizadas las medidas debe plasmar los resultados en un informe que contenga no más de 6 páginas y los siguientes puntos:

- Identificación de los estudiantes.
 - Descripción del filtro que usó para la opción 3 y cuál fue la estrategia utilizada para dividir la imagen.
 - Describa las correcciones de estilo que realizó a los programas suministrados.
 - Describa la plataforma de experimentación: sistema operativo, ambiente de ejecución (COCALC, Virtual box, etc), características del HW (o de la máquina virtual): número de núcleos de CPU, tamaño de la memoria, etc.
 - Muestre para cada experimento: problema planteado, hipótesis, descripción de los experimentos, resultados en forma gráfica y análisis de los resultados obtenidos. Las siguientes preguntas le pueden ayudar con el análisis:
 - Investigue sobre otras herramientas para medir el tiempo total de ejecución de un programa y el uso de la memoria. Mencione tales herramientas.
 - Investigue sobre otras métricas utilizadas para medir el desempeño de un programa aparte del tiempo de ejecución y/o uso de la memoria. Cómo medir, por ejemplo el uso de CPU en modo kernel o usuario, el número de interrupciones, etc.
- ¿Qué puede concluir de la solución concurrente con respecto a la secuencial? ¿Qué pasa cuando aumenta el tamaño de la imagen? ¿Qué pasa cuando aumenta el número de hilos? ¿Hay diferencia entre las soluciones secuenciales, si la respuesta es afirmativa, por qué?

Observaciones Adicionales

El proyecto lo deben realizar en grupos **de como máximo tres estudiantes**. Lo deben entregar el **(lunes 17 de abril)**. La sustentación se realizará en las horas de práctica. La entrega consiste de los códigos fuente, el makefile (todo en un archivo en formato zip) y el informe (formato pdf). Para poder sustentar, todos estos archivos deben haber sido colocados previamente en BS.

El día de la sustentación deben mostrar las imágenes que usaron para las mediciones. No las suban a la plataforma. Los profesores llevarán los casos de prueba del proyecto (con qué parámetro se ejecutarán los programas y las imágenes a usar)

En su código debe validar llamadas al sistema y parámetros de entrada. Use la guía de estilo C para codificar y comentar lo que añadan al programa para realizar la solución concurrente.

El proyecto lo deben hacer en lenguaje C (ansi C)

Nota: Cualquier duda sobre el enunciado del proyecto debe consultarla con su profesor en forma oportuna. La comprensión del problema y su correcta implementación, según lo indica el enunciado, es parte de lo que se está evaluando.

Suerte

Prof. Mariela Curiel/Ricardo Gonzalez/Osberth de Castro.