

Proyecto 2: Ordenando Números con threads

Ricardo Castro Jiménez
Carmen Hidalgo Paz
Jorge Guevara Chavarría

Escuela de Ingeniería en Computación, Tecnológico de Costa Rica, San José, Costa Rica

riccastro@estudiantec.cr
carmenhidalgopaz@estudiantec.cr
joguevara@estudiantec.cr

Se hizo un programa en C que ordena los datos utilizando hilos de ejecución (threads) y los muestra gráficamente dentro de cuatro círculos simultáneamente. El usuario puede seleccionar cuántos datos quiere ver, cómo deben estar inicialmente (ordenados o mezclados) y qué algoritmo utilizar en cada uno de los cuatro discos. Al presionar el botón de iniciar, se pueden ver los cuatro algoritmos de ordenamiento al mismo tiempo, mostrando el paso a paso cómo se reorganizan los datos en cada gráfico.

En esta versión aprendimos a trabajar con la biblioteca pthread para ejecutar múltiples algoritmos en paralelo dentro del mismo programa [1]. Cada algoritmo de ordenamiento corre en su propio hilo, permitiendo que los gráficos de los discos se actualicen al mismo tiempo. Se trabajó con GTK y Glade para diseñar una interfaz gráfica que contiene cuatro menús de selección de algoritmo (uno por disco) y un botón central de inicio.

Para ver las operaciones gráficas, se adaptaron las funciones de dibujo en Cairo para que pudieran ejecutarse independientemente en cada disco sin afectar a los demás [2]. Se añadió una estructura de datos independiente para cada hilo, permitiendo registrar el número de iteraciones y de intercambios realizados por cada algoritmo.

El BubbleSort fue agregado nuevamente como uno de los algoritmos básicos, mostrando cómo elementos adyacentes se comparan e intercambian hasta ordenar el conjunto [3].

El Cocktail Sort también se utilizó, el cual mejora ligeramente el BubbleSort al recorrer el arreglo en ambas direcciones en cada iteración [4].

Se agregó el QuickSort, un algoritmo mucho más rápido que trabaja dividiendo el arreglo alrededor de un pivote. Su ejecución en un hilo independiente permite observar su velocidad comparada con los demás algoritmos [5].

El ShellSort también fue incorporado para aprovechar sus mejoras sobre el método de inserción, usando intervalos decrecientes para reorganizar los datos de forma más eficiente [6].

Además, se agregó el Gnome Sort, que es un algoritmo poco convencional pero muy visual. Su lógica es simple: si dos elementos están desordenados, se intercambian y se retrocede una posición, como si se deshiciera el avance hasta que todo esté en orden. Aunque no es eficiente, se escogió por su comportamiento llamativo para efectos visuales [7].

Después se incorporó el Exchange Sort, un algoritmo que compara todos los pares posibles de elementos y los intercambia si están en el orden incorrecto. Aunque no es eficiente para listas grandes, su lógica directa lo hace útil para fines educativos y para ilustrar el funcionamiento de intercambios múltiples dentro de un arreglo [8].

También se agregó el Selection Sort, conocido por su simplicidad. Este algoritmo selecciona iterativamente el valor más pequeño del arreglo no ordenado y lo coloca en su posición final. No requiere intercambios innecesarios y mantiene un comportamiento predecible, aunque no mejora mucho en el mejor de los casos [9].

El Insertion Sort fue implementado por su eficacia en arreglos pequeños o parcialmente ordenados. Su lógica consiste en tomar cada elemento y colocarlo en la posición correcta dentro de la parte ya ordenada, como si se fueran insertando cartas en una mano. Es rápido en casos favorables y fácil de visualizar [10].

Asimismo, se añadió el Merge Sort, un algoritmo que divide recursivamente el arreglo en mitades, las ordena y luego las fusiona. Gracias a su estabilidad y eficiencia incluso en casos desfavorables, se utiliza comúnmente en situaciones donde se requiere rendimiento garantizado. Se implementó respetando el orden ascendente y aprovechando su naturaleza recursiva [11].

El último algoritmo que se integró fue el Pancake Sort, un algoritmo poco convencional que ordena mediante flips, como si se tratara de apilar y dar la vuelta a panqueques. Aunque su rendimiento no es competitivo frente a otros métodos modernos, fue escogido por su carácter visual y la forma en que ordena los datos usando solo flips [12].

Todos los algoritmos fueron adaptados para trabajar dentro del modelo multihilo, garantizando que cada uno pudiera correr sin interferir con el otro. También se cuidó que el programa no mostrara errores de segmentación al ejecutarse múltiples hilos simultáneamente.

Se utilizó el algoritmo de Fisher-Yates para barajar los datos de forma aleatoria antes de iniciar cada ordenamiento, asegurando que los cuatro algoritmos partieran de los mismos datos [13].

Además, se mejoraron los tiempos de visualización usando delays controlados, para que los cambios en los gráficos fueran perceptibles para el usuario, pero sin detener el funcionamiento general del programa.

Finalmente, se reforzó la robustez del proyecto eliminando salidas a consola que pudieran afectar la presentación gráfica, como se requería para una entrega de alta calidad.

FUENTES CONSULTADAS:

- [1] Tutorialspoint, “C - Multithreading”, [En línea]. Disponible en:
https://www.tutorialspoint.com/cprogramming/c_multithreading.htm [Accesado: 20 de abril del 2025].
- [2] K. O’Kane, “Linux Gtk Glade Programming Part 1”, YouTube, 22 de mar. del 2019.
[video en línea]. Disponible en:
https://www.youtube.com/watch?v=g-KDOH_uqPk&list=PLmMgHNtOIstZEvqYJncYUx52n8_OV0uWy&index=2 [Accesado: 8 de mar. del 2025].
- [3] GeeksforGeeks, “Bubble Sort”, [En línea]. Disponible en:
<https://www.geeksforgeeks.org/bubble-sort/> [Accesado: 6 de abril del 2025].
- [4] GeeksforGeeks, “Cocktail Sort”, [En línea]. Disponible en:
<https://www.geeksforgeeks.org/cocktail-sort/> [Accesado: 6 de abril del 2025].
- [5] GeeksforGeeks, “QuickSort – Data Structure and Algorithm Tutorials”, [En línea].
Disponible en: <https://www.geeksforgeeks.org/quick-sort/> [Accesado: 6 de abril del 2025].
- [6] GeeksforGeeks, “ShellSort – Data Structure and Algorithm Tutorials”, [En línea].
Disponible en: <https://www.geeksforgeeks.org/shellsort/> [Accesado: 6 de abril del 2025].
- [7] GeeksforGeeks, “Gnome Sort – Data Structure and Algorithm Tutorials”, [En línea].
Disponible en: <https://www.geeksforgeeks.org/gnome-sort-a-stupid-one/> [Accesado: 6 de abril del 2025].
- [8] GeeksforGeeks, “Introduction to Exchange Sort Algorithm”, [En línea]. Disponible en:
<https://www.geeksforgeeks.org/introduction-to-exchange-sort-algorithm/>
- [9] GeeksforGeeks, “Selection Sort”, [En línea]. Disponible en:
https://www.geeksforgeeks.org/selection-sort-algorithm-2/?ref=header_outind
- [10] GeeksforGeeks, “Insertion Sort Algorithm”, [En línea]. Disponible en:
https://www.geeksforgeeks.org/insertion-sort-algorithm/?ref=header_outind
- [11] GeeksforGeeks, “Merge Sort – Data Structure and Algorithms Tutorials”, [En línea].
Disponible en: https://www.geeksforgeeks.org/merge-sort/?ref=header_outind
- [12] GeeksforGeeks, “A Pancake Sorting Problem”, [En línea]. Disponible en:
https://www.geeksforgeeks.org/a-pancake-sorting-question/?ref=header_outind
- [13] Inside code, “How to shuffle an array (Fisher-Yates algorithm) - Inside code”, YouTube,
18 de oct. del 2020. [video en línea]. Disponible en:
<https://www.youtube.com/watch?v=4zx5bM2OcvA&t=57s> [Accesado: 25 de feb. del 2025].