Fabián Andrés Merchán Jiménez

David Stiven Ávila González

Primer proyecto: Sort concurrente

Descripción del problema planteado

Para este proyecto se desea observar, analizar y comprobar la teoría de la optimización del rendimiento en la ejecución de tareas concurrentes a través del uso de hilos y procesos vistos en clase. Para ello se recurrirá al manejo de archivos de texto, los cuales se ordenarán de manera concurrente, usando tanto hilos como procesos y luego se hará una comparación entre los tiempos destinados por cada programa para realizar la ejecución.

Algoritmos de ordenamiento implementados

Para esta entrega se implementa el algoritmo de ordenamiento de burbuja cuya complejidad es , sin embargo esta complejidad puede cambiar según la cantidad de datos y el orden de los mismos, lo que es una interesante comparación con el “mergeSort” que implementa Linux en su terminal de comandos.

“MergeSort” es un algoritmo de ordenamiento que se basa en la división de los datos y la recursividad para su fin. La división de los datos le permite optimizar y organizar el espacio en memoria, y así ordenarlos con facilidad. La complejidad de este algoritmo es lo que quiere decir que es mucho más rápido que el ordenamiento de burbuja. Sin embargo es eso lo que se quiere demostrar en este proyecto.

Llamada al sistema exec

Tablas de resultados

|  |  |  |  |
| --- | --- | --- | --- |
| **Algoritmo** | **Cantidad de lineas** | | |
| **100** | **500** | **1000** |
| **csortp** | Real: 172 ms  User: 4 ms  Sys: 84 ms | Real: 362 ms  User: 36 ms  sys: 148 ms | Real: 538 ms  User: 136 ms  sys: 216 ms |
| **csortpexec** | Real: 121 ms  User: 23 ms  Sys: 28 ms | Real: 105 ms  User: 8 ms  Sys: 36 ms | Real: 143ms  User: 12 ms  Sys: 52 ms |
| **csorth** | Real: 199 ms  User: 8 ms  Sys: 100 ms | Real: 251 ms  User: 40 ms  Sys: 132 ms | Real: 440 ms  User: 152 ms  Sys: 204 ms |

Gráficas obtenidas

Análisis de resultados