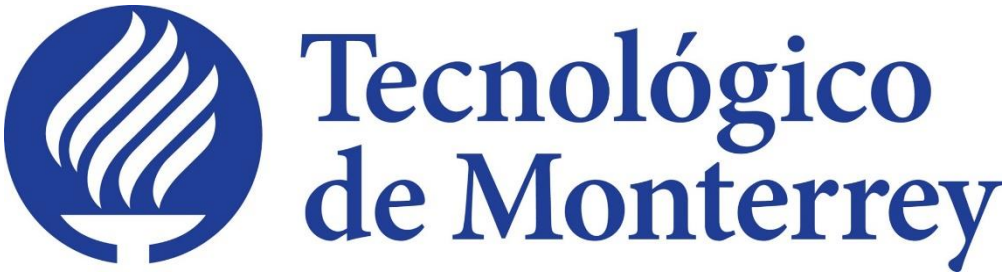


Actividad Integradora 5.3 Resaltador de sintaxis paralelo

Nombre: Diego Sú Gómez

Matrícula: A01620476

Materia: Implementación de métodos computacionales

Semestre: Enero – junio 2022

Fecha de entrega: lunes 7 de junio de 2022

Profesores: Luis Ricardo Peña Llamas

Actividad Integradora 5.3 Resaltador de sintaxis paralelo

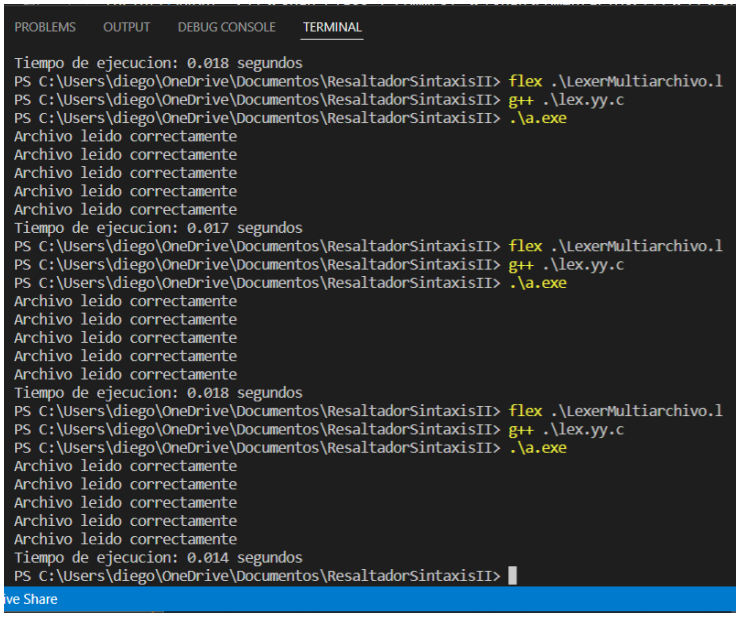
Para la segunda actividad integradora del semestre, se tuvieron que realizar algunas modificaciones al resaltador de sintaxis. La primera de ellas fue que el resaltador tenía que ser capaz de procesar varios archivos a la vez, y devolverlos en distintas páginas HTML. Para esto se implementaron varias modificaciones con respecto al programa original. Entre ellas, un sistema para extraer todos los archivos de una ubicación en específico, para después asignar esos nombres de los archivos a los HTML que mostrarían los resultados. Además de implementar distintas filas para poder almacenar esta información, y después pasar la información a los archivos y al lex en sí, haciendo que todos los archivos que estuvieran en un fólder fuesen leídos y analizados uno a uno y los resultados saldrían cada uno en un archivo diferente. Sin embargo, este programa también fue realizado de manera secuencial, por lo que aunque era más efectivo, la eficiencia no iba a ser la mejor de todas.

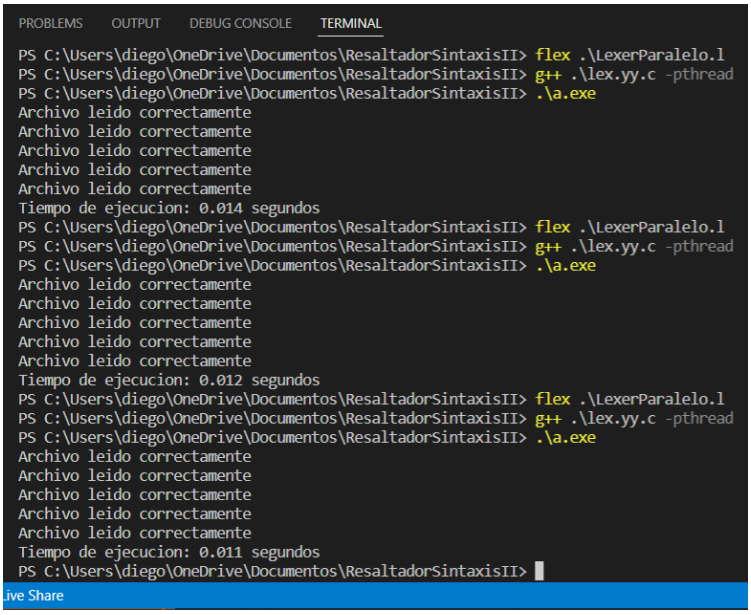
La segunda modificación que se hizo fue hacer otra versión que realizara exactamente lo mismo que la versión de multi archivos del resaltador, pero utilizando la programación paralela, haciendo uso de hilos para que cada uno pudiera analizar un archivo a la vez, haciendo que el tiempo de ejecución fuese menor, en teoría, y que se aprovecharan de mejor forma los recursos de la computadora. Para esto, se utilizó la técnica de threading, en donde cada uno de los hilos se encargaba de un archivo por separado, permitiendo que las tareas se pudieran dividir adecuadamente.

Tras haber concluido con la realización de ambos programas, se procedió a hacer un análisis acerca de las complejidades, los tiempos de ejecución y los algoritmos implementados. Con esta información, y ambos programas ya realizados, se midieron tres tiempos de ejecución de cada uno de los programas. Los resultados se muestran a continuación en la siguiente tabla.

Programa	Ejecución 1	Ejecución 2	Ejecución 3
Secuencial Multi archivo	0.017 segundos	0.018 segundos	0.014 segundos
Paralelo	0.014 segundos	0.012 segundos	0.011 segundos

Actividad Integradora 5.3 Resaltador de sintaxis paralelo
Diego Sú Gómez – A01620476





Cómo se puede observar tanto en la tabla, como en las imágenes que se muestran en la parte de arriba de la página, los tiempos del resaltador de sintaxis paralelo fueron menores. Si bien es cierto que no fueron significativamente menores, se puede observar que la programación paralela es muy útil para este tipo de escenarios. Estas diferencias se verían mucho más marcadas si se usasen archivos más pesados o una mayor cantidad de archivos, haciendo evidente que en este tipo de casos la programación paralela es más útil. A continuación, se calculará el speed up que se obtuvo con las medidas.

$$SpeedUp = \frac{0.017}{0.014} = 1.21$$

$$SpeedUp = \frac{0.018}{0.012} = 1.5$$

$$SpeedUp = \frac{0.014}{0.011} = 1.27$$

Cómo se puede observar al calcular el speed up, el programa realizado en paralelo es mucho más rápido que el secuencial, llegando a ser aproximadamente 1.3/1.4 veces más rápido. Esto confirma lo dicho previamente, que en escenarios más complejos, con más archivos o con archivos bastante más pesados, el resaltador hecho en paralelo podría desempeñarse bastante mejor que uno hecho de forma secuencial. Esto también se puede comprobar, además de con los tiempos de ejecución y el speed up, con la complejidad temporal de los algoritmos utilizados. El programa secuencial tiene una complejidad cuadrática de O(n^2), mientras que el programa paralelo tiene una complejidad lineal O(n), debido a que únicamente utiliza una iteración para guardar las direcciones y los nombres de los archivos de texto. Esto mismo permite comprobar y fundamentar el porqué es que los programas

hechos de manera secuencial no siempre son los más útiles para resolver problemas, y que hay bastantes escenarios en donde implementar distintos tipos de programación puede llegar a ser mucho más útil y funcional.

Este proyecto me ayudó a desarrollar distintas habilidades computacionales y de aprendizaje, además de que también fue bastante útil para poder reconocer la capacidad que tienen este tipo de tecnologías, junto con su alcance y las repercusiones que puede llegar a tener en la sociedad si no se les da un buen uso. Este tipo de tecnologías requieren que los programadores y desarrolladores tengan siempre muy presente los valores éticos y morales para evitar que estas tecnologías y algoritmos caigan en manos equivocadas. Si esto llegase a pasar, algoritmos de este estilo se pueden llegar a usar para vulnerar la información personal de la gente y revelar contraseñas, cuentas bancarias e información personal que podría llegar a causar problemas mayores. Además de eso, si se llegase a tener intenciones equivocadas con este tipo de softwares, se podría despertar el interés de organizaciones o personas que se interesarían en darle mal uso a estos algoritmos, por lo que es necesario el conocer el alcance que pueden llegar a tener este tipo de algoritmos, y ser muy conscientes y responsables tanto de manera personal, como con la sociedad, al no darle malos usos o ser egoístas y ver primero por el bienestar personal antes de la seguridad o el bienestar social. Este curso me permitió poder entender la importancia de la ética y la responsabilidad al momento de desarrollar software, además de poder conocer a fondo los distintos tipos de programación, los lenguajes y gramáticas y cómo estos son bastante útiles para poder realizar y resolver distintos problemas.