Tecnológico de Monterrey

“Actividad Integradora 5.3 - Resaltador de Sintaxis Paralelo "

Implementación de Métodos Computacionales(Gpo 3)

Profesor:
Pedro Oscar Perez Murueta

Alumno:

Ricardo Andrés Cáceres Villibord A01706972

25 de Julio del 2022

Para este proyecto, se ha tomado en cuenta el lenguaje de C++, para poder analizar los elementos léxicos del lenguaje, y así poder crear un analizador léxico en C++ donde su objetivo sea poder resaltar la sintaxis del lenguaje. El analizador léxico tiene como principal tarea leer los caracteres de entrada de un programa fuente (en este caso un archivo de texto escrito en el lenguaje C++), y producir una secuencia de de tokens para cada lexema. El token consiste en el token y un atributo especial, que en este caso será el tag de html que se le asignará.

Las categorías identificadas fueron las siguientes:

Palabras Reservadas
namespace|if|else|for|while|do|switch|case|continue|break|int|float|double|long|string|cin|cout|char|const|void|bool|false|true|static|virtual|new|class|private|public|throw|this|catch|try|return|volatile|extern|unio n|sizeof|pair|printf|scanf

Operadores
+, ++, -, –, !, <, > , <<, >>, <=, >=, =, ==, &, *, /, ^, ?:, +=, -=, <<=, >>=, &=, |=, ^=, *=, /=, |, ||, &&

Identificadores
Son los nombres que se le proporcionan a las variables.

Includes
Son librerías que se incluyen en el documento para usar en el código. Están formadas de la siguiente forma: #include <LIBRERIA>

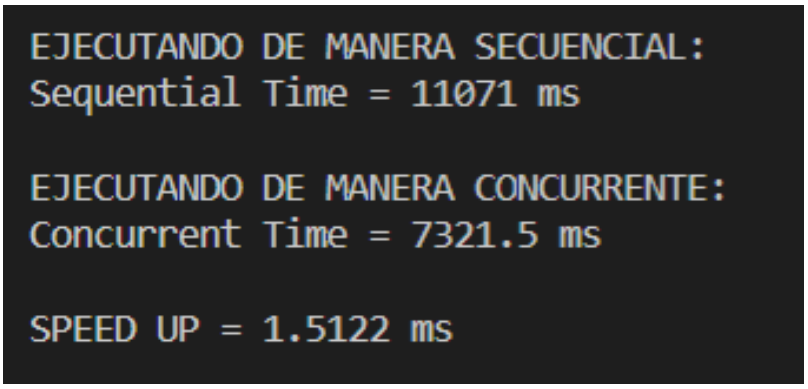
Números
Números enteros y flotantes

Comentarios
Oraciones en donde se empieza con //.

Variables
Palabras usadas como identificadores.

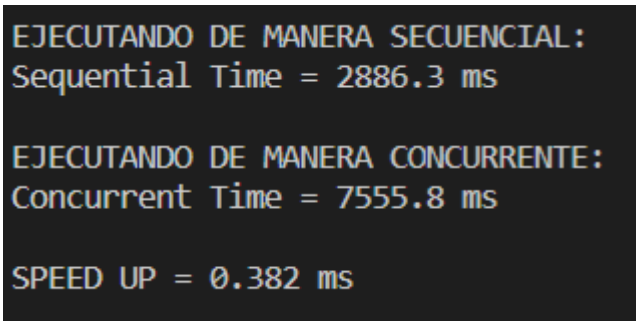
Medición de Tiempo de Ejecución

- **Con 8 archivos y 8 hilos de trabajo:**



Podemos observar que al utilizar 8 hilos de trabajo en 8 archivos, el proceso se puede llegar a hacer 1.5 veces más rápido que en comparación de hacerlo de manera secuencial.

- **Con 4 archivos y 8 hilos de trabajo:**



Podemos observar que al utilizar muy pocos archivos, no ocurre ningún speedup. De hecho es más lento utilizar multihilos, ya que se tarda más en el proceso de crear los hilos. Y es por esa razón que el secuencial termina siendo más rápido.

Conclusiones

Al haber realizado el resaltador de sintaxis para el lenguaje C++, se ha utilizado herramientas importantes como lo son las expresiones regulares. Las expresiones regulares han sido ocupadas para poder categorizar cada token y así poder asignarle un color en el HTML. Se hicieron múltiples funciones booleanas las cuales se encargaban de revisar a qué categoría pertenecia cada token (el cual podría ser una palabra completa o solamente podía ser un carácter) y así poder asignarle su respectiva tag de HTML y agregarla dentro de un archivo de salida correspondiente al archivo que se estaba leyendo.

Se hizo una comparación con la ejecución del programa siendo implementado de manera secuencial y siendo implementado de manera concurrente. Y en base a las pruebas ejecutadas, podemos observar como al ejecutar el programa de manera concurrente, es decir utilizando multihilos, podemos disminuir el tiempo de ejecución ya que esto hace que dividamos la tarea en varios procesos/hilos de trabajo en nuestro procesador.

También podemos resaltar el análisis de complejidad para nuestras funciones. Las funciones que revisaban si el token pertenecía a una categoría son de O(1), y que estas solo entran a la función y checan una vez. Mientras que nuestra función que crea el vector de palabras recorre n*n pasos en el peor de los casos, por lo que su complejidad temporal es de O(n^2). Y finalmente nuestra función principal que construye los HTML es de O(n^a) en el peor de los casos.

De este proyecto también podemos destacar algunas implicaciones éticas, y es una tecnología que puede llegar a ser de mucha utilidad para la sociedad. Una de las utilidades es para las personas que no tienen las capacidades visuales adecuadas. Al poder resaltar de diferentes colores los diferentes lexemas del lenguaje de programación, estas personas pueden identificar y diferenciar más rápido a qué categoría pertenece cada token. Y de la misma manera, esta es una solución muy versátil ya que los colores se pueden ajustar para las necesidades de cada persona.

Link Repositorio Github:
https://github.com/Caceres-A01706972/TC2037/tree/main/Situación%20Problema/Resaltador%20de%20Sintaxis%202