

**Instituto Tecnológico y de Estudios
Superiores de Monterrey**

Campus Monterrey



Implementación de Métodos Computacionales

Actividad Integradora 5.3 Resaltador de sintaxis paralelo

Alejandro Hernández Carrales | A01721287

TC2037.570

Pedro Oscar Pérez Murueta

Julio 25, 2022

Para esta entrega, se mantuvo la solución anterior al problema del resaltador de sintaxis en Racket. Sin embargo, el cambio de lenguajes de Racket a c++ me hizo tener que cambiar la forma de leer y lidiar con archivos. Como antes no era necesario mandar como parámetros del código las direcciones de los archivos, tuve que aprender a utilizar argc y argv como apuntador. Ya que ahora tendría que lidiar con la administración de los hilos del procesador, decidí acercarme a la solución dividiendo los archivos por cada hilo. Esto significa que, dado el número de archivos, los hilos recibirían una fracción de estos y leerían independientemente los archivos, haciendo que estos salgan como vayan terminando.

Debido a que antes solo leía el archivo y guardaba en un string en racket, ahora tengo que leer por línea cada archivo y agregárselo a "code". Esto significa que el programa leería cada n número de líneas por cada n número de archivos. Esto aumentó la complejidad de mi código ya que ahora el programa iniciaría con un ciclo for para limpiar los nombres de los archivos seguido de otro ciclo for por el número de archivos y dentro de este un ciclo do-while $O(n)$, haciendo que la complejidad en notación big O sea $O(n^2)$, haciendo que se tarde más tiempo dependiendo de la cantidad y el tamaño de los archivos a resaltar.

Con unas mejoras a la calidad de vida del resaltador como el cambio de colores, la exportación a un folder organizado que mantenga el nombre de los archivos input y con también unos compromisos que se tuvieron que tomar como aumentar la complejidad del archivo y que los comentarios no sean 100% correctamente resaltados por la manera de acercarse a la solución, esta nueva versión del resaltador ahora funciona en Racket y depende de Regex para poder identificar las palabras reservadas.

Cabe mencionar que haber escrito el resaltador en c++ abrió la posibilidad para que se pudiera comparara su implementación secuencial con la paralela, tomando ventaja de los hilos disponibles del procesador, mostrando en las pruebas que el "speed up" es de hasta un 2.3 veces en el concurrente, compensando bastante por el aumento en complejidad. Con estas medidas utilizadas, se ha hecho una notable mejora en el resaltador.

Como conclusión, es importante tomar en consideración las implicaciones éticas que proyectos como este pueden tener a la hora de ser compartidos. Una persona con acceso a los métodos vistos hoy puede utilizar las reglas de regex para leer código confidencial de una manera extremadamente rápida. Las herramientas vistas durante el proyecto deben ser tomadas con seriedad y precaución ya que además de vulnerabilidades de seguridad que nuestro programa no toma en cuenta, puede llegar a invadir la privacidad o documentos confidenciales que potencialmente pueden llegar a nosotros por medio de nuestro código. Programar responsablemente es una de las primeras lecciones que nos enseñan y debería ser utilizada cada vez que se haga un proyecto.