



Tecnológico de Monterrey

Instituto Tecnológico y de Estudios Superiores Monterrey
Campus Estado de México

Materia: Lenguajes de programación y traductores
Semestre Ago - Dic 2018
8 de Octubre de 2018

Diego Canizales Bollain Goytia A01376119
Isain Cuadra Rivas A01375997
Arturo Velázquez

Profesor: Ingeniero Edgar Vallejo Clemente

Diego:

A mi me tocó trabajar con los dos programas, tanto el léxico (flex) como la gramática (bison). Mientras estaba escribiendo los dos programas me encontré con varios problemas que no entendía muy bien por qué sucedían pero me dio gusto saber que al final se solucionaran. Primero no sabíamos muy bien cómo implementar el léxico, desde cómo decirle a nuestro scanner que entendiera todas las palabras claves de C, hasta implementar algunas expresiones regulares que pudieran entender los valores primitivos. También me tocó trabajar en la implementación del parser, en el que se dedica a encontrar la gramática del programa que está recibiendo. Implementé en el lex los valores del espacio en blanco, comentarios, que detectara los keywords y la definición del error, así como en el parser el cómo iba a recibir estos valores en valores terminales y la implementación del error que despliega cuando hay un error en la sintaxis. Arturo me ayudó mucho a entender cómo funciona en realidad el compilador y cómo se deben implementar las reglas, ya que muchos de los tutoriales que encontramos estaban escritos en c++ y pues él es un excelente programador. También me enseñó a utilizar git para el control de versiones, lo que nos facilitó mucho la vida. Isaín también es un excelente programador y él me ayudó a entender mucho la lógica de cómo es que Bison lee el programa, lo que contribuyó en poder arreglar la funcionalidad del mismo.

Isain:

Para esta entrega estuve trabajando en la parte del Bison y del SymbolTable, mis principales participaciones para esta entrega fue encontrar la lógica que asocia un identificador con un keyword, identificar un método declarado, identificar todos los elementos de una variable que ha sido declarada, la identificación de números hexadecimales. Este trabajo ha sido increíblemente divertido ya que he puesto en práctica distintos conocimientos, como los REGEX, el concepto de Token, programación en C, como esta compuesto C, tambien he puesto en práctica todo lo visto en clase acerca de Flex and Bison. El aprendizaje que me deja el desarrollo de este proyecto es el cómo se encuentra desarrollado un lenguaje de programación internamente, y es que es increíble como te un archivo de texto el parser y el scanner trabajan en conjunto para poder identificar las instrucciones que se especifican y después poder hacer magia con ello. Externo al desarrollo de este proyecto aprendí a utilizar contenedores de docker ya que en lugar de hacer una partición o crear una máquina virtual decidí hacerlo dentro de un contenedor de Docker ya que es una tecnología que está acaparando mucho mercado hoy en día y se que eventualmente tendría que aprender a utilizarla, ¡qué mejor que ahorita que soy estudiante!.

Arturo:

Para esta entrega tuve que participar en la parte del repositorio, la definición de los diferentes tokens en flex y las reglas de la gramática para el lenguaje. Lo que me intereso son los diferentes procesos que existen para poder interpretar en lenguaje de programación. Primero tenemos que encontrar todos los diferentes tokens que pueden existir en el programa para poder evaluarlos después en la sintaxis para el lenguaje. Basado en los

tokens que creamos desde flex ahora podemos importar todos estos valores a Bison. Bison es donde tenemos que definir si lo que tenemos en nuestro archivo es válido. Esto se desarrolla con las diferentes reglas que generamos dentro del archivo de Bison. Una parte importante de las reglas es las prioridades que existen dentro de este archivo. Entonces tenemos que declarar como dependiendo de los tokens generan ciertas reglas en bison. Podemos separar cada regla por secciones y dentro de cada sección podemos tener reglas sobre expresiones esperadas en esa sección. Isain entendió muy bien como se puede programar para poder identificar las diferentes variables y cómo poder visualizar los diferentes tokens. Diego me ayudo mucho en poder desarrollar las diferentes reglas para los tokens. Con la combinación de ambas perspectivas podemos conseguir en desarrollar el proyecto.

```
→ Flex-Bison git:(master) X ./clegacy ./scripts/helloworld.c
The file evaluated is: ./scripts/helloworld.c
Bison found a special symbol: 35
Bison found a string: include
Bison found a special symbol: 60
Bison found a string: stdio
Bison found a special symbol: 46
Bison found a string: h
Bison found a special symbol: 62
Bison found a method: main
Bison found a special symbol: 123
Bison found a variable definition: int 14
Bison found a special symbol: 59
Bison found a string: a
Bison found a special symbol: 61
Bison found an int: 123412
Bison found a special symbol: 59
Bison found a string: printf
Bison found a special symbol: 40
Bison found a special symbol: 34
Bison found a string: Hello
Bison found a special symbol: 44
Bison found a string: World
Bison found a special symbol: 33
Bison found a special symbol: 34
Bison found a special symbol: 41
Bison found a special symbol: 59
Bison found a keyword: return
Bison found an int: 0
Bison found a special symbol: 59
Bison found a special symbol: 125
```