Documentación de proyecto.

Compilador

Fecha: [20/09/2019]

Tabla de contenido

1. Introducción

1.1 Propósito.

Construir un compilador de pocas instrucciones del lenguaje ‘C’, utilizando las técnicas y procesos que se mencionan en los requerimientos, para así llegar a un compilador, que, aunque no soporte todo el lenguaje esté hecho de tal forma que sea sencillo el poder agregar más instrucciones sin demasiado esfuerzo y además siga siendo óptimo.

2. Requerimientos

El proyecto deberá cumplir los siguientes puntos:

- Deberá estar modulado en Scanner, Parser y Generador de código.
- De forma opcional deberá incluirse un optimizador.
- Deberá funcionar en un entorno Posix Unix.
- Todas las entregas deberán ser en la plataforma Github.
- El compilador deberá reconocer la sintaxis para devolver un entero.

3. Plan de trabajo

Nombre y Apellido	Cargo
Alberto Alonzo Lona López	Project Manager
Jair de Jesús Gallegos Santiago	Integrador
De Jesús Celestino Irving Manuel	Tester
Ernesto Palacio	Arquitecto

- Aprender sintaxis de Elixir

Tiempo estimado de trabajo efectivo: 8hrs

Aprender y familiarizarse con la sintaxis de elixir.

- Planeación y elaboración Limpiador

Tiempo estimado de trabajo efectivo: 6hrs

Con los conceptos vistos en clase planificar y elaborar el limpiador

- Planeación y elaboración Lexer

Tiempo estimado de trabajo efectivo: 6hrs

Con los conceptos vistos en clase planificar y elaborar el scanner/lexer

- Planeación y elaboración Parser
  - Parse_program→Parse_function→Parse_statement→Parse_expression
  - AST

Tiempo estimado de trabajo efectivo: 8hrs

Con los conceptos vistos en clase planificar y elaborar el parser para validar la sintaxis correcta y generar el árbol AST

- Elaboración Generador de código

Tiempo estimado de trabajo efectivo: 5hrs

Con los conceptos vistos en clase planificar y elaborar el generador de código

- Elaboración Optimizador (Opcional)

Tiempo estimado de trabajo efectivo: 5hrs

Optimización de los puntos anteriores.

- Extras

  Tiempo estimado de trabajo efectivo: 5hrs

  Extras que agregar:

  Señalización e indicación de errores

4. Arquitectura del compilador

4.1 Entorno operativo

Este compilador está diseñado para el entorno POSIX, Linux.

POSIX es un acrónimo de «Portable Operating System Interface», o también Interfaz Portable del Sistema Operativo. El shell de POSIX se basa en el estándar de POSIX IEEE P1003.2, que es un conjunto de normas codificadas por la entidad IEEE y emitido por las entidades ISO y ANSI.

5. Plan de pruebas

Las pruebas serán las que están establecidas en el repositorio de Nora Sandler, serán 12 pruebas en total, divididas en 6 pruebas cuyos resultados se esperan sean inválidos y 6 más que se esperan sean válidos.

5.1 Pruebas

Nombre de la prueba	Descripción	Resultado esperado	Resultado obtenido
Missing Paren	En la función *'main'* no estará el paréntesis de cierre.	Inválido.	Inválido
Missing retval	Después de la sentencia *'return'* no estará ningún valor de regreso.	Inválido.	Inválido
No brace	No estará la llave de cierre.	Inválido.	Inválido

No semicolon	Después de la función *'main'* no estarán ninguno de los paréntesis.	Inválido.	Inválido
No space	Después de la sentencia *'return'* no habrá ningún espacio entre él y el entero.	Inválido.	Inválido
Wrong case	La sentencia *'return'* estará escrita en mayúsculas.	Inválido.	Inválido
Multi digit	El valor entero de regreso está formado por más de un dígito.	Válido.	Válido.
New lines	Cada una de las sentencias que forman el programa estarán separadas por un salto de línea.	Válido.	Válido.
No new lines	Todo el código del programa estará en una sola línea, pero respetando la sintaxis.	Válido.	Válido.
Return 0.	El valor de regreso será un número cero.	Válido.	Válido.
Return 2	El valor de regreso será un número dos.	Válido.	Válido.
Spaces	Cada una de las sentencias estará separada por espacios.	Válido.	Válido.

6. Conclusiones.

Si bien se presentaron dificultades desde el inicio del proyecto debido a que uno como administrador del proyecto apenas se encuentra cursando la asignatura en la cual se enseñan todos estos métodos para llevar a cabo la administración de un proyecto, considero que fue una buena oportunidad de llevar a la práctica los conceptos y métodos aprendidos en la asignatura. Si bien todavía hay bastantes cosas que debo mejorar tanto como PM, así como que deben mejorar en la interacción a con el equipo, sin embargo, se logró llegar a una solución bastante cercana a la establecida en el propósito inicial, a pesar de las dificultades y los malentendidos del equipo.

En este proyecto no sólo aprendí a hacer un compilador, creo que el trabajo en equipo fue la mejor experiencia que me dejó este proyecto, además aprendí a usar un nuevo lenguaje y a dividir responsabilidades entre módulos de trabajo en conjunto, así como el aplicar ciertas metodologías que complementan a lo visto en materias previas de la carrera.

7. Referencias.

- https://www.neoguias.com/que-es-el-shell-de-posix/
- https://github.com/nlsandler/write_a_c_compiler
- https://norasandler.com/2017/11/29/Write-a-Compiler.html
- https://llvm.org/
- https://github.com/hiphoox/nqcc_elixir