

Universidad Nacional Autónoma de México
Facultad de Ingeniería
División de Ingeniería Eléctrica (DIE)

Asignatura: Compiladores

Entrega 1 Compilador de C

Integrantes:

Guadarrama Ortega César Alejandro
Villa Vega Juan Darío

Grupo: 4

Fecha de Entrega: 17 de Marzo del 2020

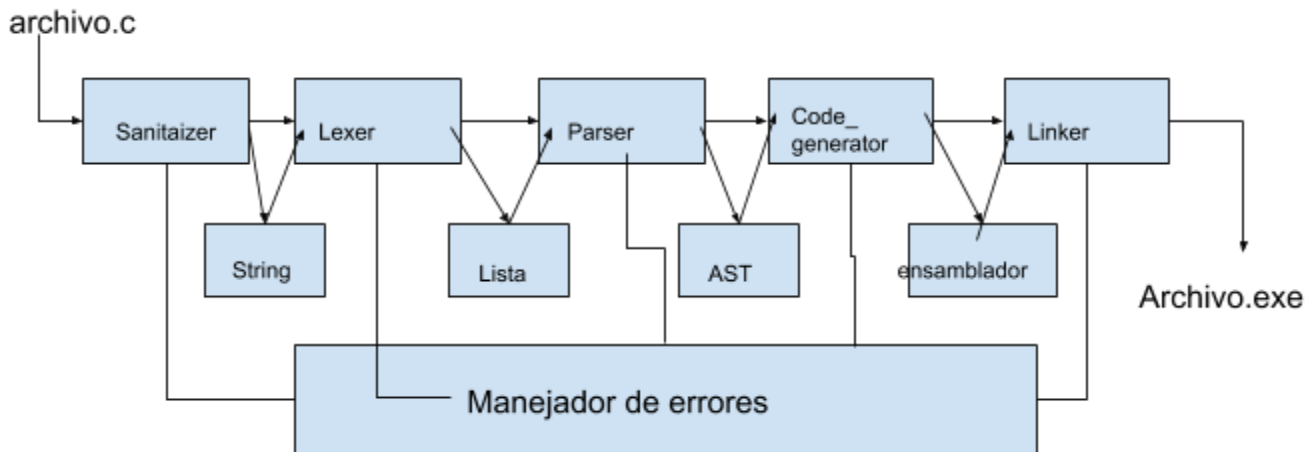
Introducción

El campo de los compiladores es muy grande, aunque es uno de los campos menos aprovechados por algunas instituciones académicas es interesante ver cómo es que funciona uno. Para este proyecto cada uno de los miembros del equipo de trabajo tiene una tarea asignada que se debe respetar aunque esto también depende de ciertas circunstancias. Le realizará un pre-compilador de C con el lenguaje de programación de paradigma funcional llamado Elixir. La planeación, requerimientos y arquitectura de nuestro programa se mostrará a continuación.

Descripción global del producto

El producto a presentar es un pseudo-compilador de lenguaje C que generará directamente un archivo ejecutable de terminación . Exe, el compilador se realizará en el lenguaje de programación de elixir.

Funciones



El diagrama del compilador (ilustración 1) nos muestra el proceso que se llevará a cabo al archivo con terminación .C, el archivo se le es entregado al módulo Sanitaizer, el cual lo convertirá en una cadena y dicha cadena será la entrada del módulo lexer, lexer convertirá la cadena en una lista de tuplas, que será entregada al parcer, el parcer lo convertirá en un Abstract Syntax Tree o AST para abreviar, el cual pasará a ser procesado por el Code_generator convirtiéndolo en un archivo ensamblador y, finalmente, el linker transformará el archivo ensamblador en un ejecutable.

Todos los módulos del programa estarán trabajando con un manejador de errores por aquellas situaciones que se puedan presentar a lo largo del procedimiento.

Características de los usuarios

El compilador está diseñado para que cualquier persona con conocimientos básicos en el uso del shell de su sistema operativo para ejecutar el compilador.

Restricciones

La computadora debe ser al menos de 64 bits

El programa solo se podrá ejecutar desde la barra de comandos(shell)

EL sistema se diseñará en un sistema pipeline

Condiciones y dependencias

La computadora en la que se ejecute debe de tener la máquina virtual de Erlang

La computadora debe de tener el compilador gcc para que funcione correctamente

Requerimientos específicos.

Requisitos funcionales.

1.-El compilador se le entregará un archivo con terminación .c y lo convertirá en un archivo .exe

2.-Dicho compilador deberá de momento retornar un número entero, es decir , tendrá que seguir la siguiente estructura:

```
int main(){  
    return 2;  
}
```

donde lo único que irá cambiando es el número entero después del return, según lo especificado en el blog de Nora Sandler primera semana

3.-El compilador deberá de soportar espacios en blanco así como saltos de linea sin que afecte el resultado.

4.-El compilador tendrá la opción del uso de banderas:

Bandera	función
--h	Muestra un menú de ayuda con toda la información contenida en esta tabla
--a	Muestra en la terminal el código ensamblador
--t	Imprimirá el árbol AST
--s	Mostrará la salida del sanitizer
--l	Mostrará la salida del lexer

Cabe destacar que puede haber más de una bandera como argumento del programa.

5.- Si hay un error en el archivo .c, el compilador deberá de reportar dicho error y la línea que está mal.

Requisitos no funcionales

El compilador se realizará en 4 partes, estas partes serán creadas guiadas a partir del blog de Nora Sandler, respectivamente las primeras cuatro semanas.

De momento la primera entrega se define para el 17 de Marzo del 2020

No es necesario el uso de interfaz gráfica.

Apéndice

Blog guía: <https://norasandler.com/>