



UNIVERSIDAD AUTONOMA DE SINALOA
FACULTAD DE INGENIERIA MOCHIS
LICENCIATURA EN INGENIERIA DE SOFTWARE



Software de sistemas

**Manual de instrucciones y declaración de sintaxis del
compilador**

Prof. Eduardo Cuhactemoc paes Palafox

Alumnos:

- Quintana Castorena Jose Eduardo
- Valdez Ornelas Pablo Antonio

Grupo: 4-02

10 de enero del 2024

El Código del compilador este hecho en el lenguaje de C solo consta de guardar en un archivo de texto en la carpeta donde esta guardado el Código y objeto del compilador para que encuentre fácilmente el archivo de texto, dicho archivo es donde puede crear un Código simple para probar el compilador.

En la misma carpeta donde envíe el compilador ahí tengo archivo de texto de ejemplos por si prueba el compilador.

Estructura del Código

Asignación de variables

- Char = cadena
- Int = entero
- Float = decimal
- Bool = booleano
- Enum = constante

Declaración de librerías

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <string.h>
#include <ctype.h>
```

Tipos de declaración de instrucción y estructuras

- Hacer
- Cout = mostrar
- Cin = leer
- If = si
- While = mientras
- Else = sino
- Elseif = sisino
- For = para

```
FILE *archivo;
char caracter;
```

```

int fila = 0;
int columna = 0;

char *listaPalRes[] = {"inicio", "cadena", "entero", "decimal", "booleano",
"constante", "hacer", "mostrar", "leer", "si", "mientras", "sino", "sisino", "para"};

struct Token { ... };
struct Nodo { ... };

struct Nodo *raiz;
struct Nodo *actual;
struct Nodo *aux;

```

Aquí se declaran las variables globales, las palabras reservadas, y se definen las estructuras Token y Nodo utilizadas para construir una lista enlazada que representa la secuencia de tokens.

Funciones de autómatas

- AutomataID
- AutomataDigit
- AutomataSimlg,
- AutomataOpeRel
- AutomataOpeLog
- AutomataOpeAri
- AutomataSim
- AutomataCadText
- AutomataComentarios

Son funciones que implementan autómatas finitos para reconocer diferentes tipos de tokens en el código fuente.

Funciones de Analizador sintáctico

- SyntaxError
- SyntaxTypeError
- AvanzarToken
- Match
- MatchTipo
- MatchTipoDato
- Sentencia
- Sent_while

- Sent_asig
- Sent_if
- Sent_for
- Sent_leer
- Sent_mostrar
- Exp
- Exp_s
- Exp_simple
- Term
- Factor

En el Código estas funciones implementan el analizador sintáctico del compilador.

Funciones de manejo de la Lista enlazada

- Insertar
- ImprimirLista

Aquí estas funciones que declare se encargan de insertar nuevos nodos en la lista enlazada y de imprimir la lista, respectivamente.

Función “main”

Esta es la función principal donde se inicializan algunas variables, se abre el archivo de código fuente y se inicia el análisis léxico y sintáctico la función main también imprime "Compilación con éxito" si no se encuentran errores.

Flujo principal

- Se utiliza un bucle principal para leer caracteres del archivo de código fuente.
- Los autómatas finitos se utilizan para reconocer diferentes tipos de tokens (identificadores, números, operadores, cadenas, etc.).
- El analizador sintáctico se encarga de verificar que la secuencia de tokens cumpla con la gramática del lenguaje.

Ejemplo al iniciar una instrucción

Inicio {.....}

Ejemplo de un código de nuestro compilador como iniciar y como declarar las instrucciones en pseudocódigo

```
inicio{  
    entero a = 5;  
    entero b = 7;  
    entero resultado = 0;  
  
    resultado = a + b;  
    mostrar(resultado);  
}
```