



Universidad Nacional Autónoma de México

Facultad de Ingeniería



Asignatura: Compiladores

Alumnos:

Carrillo Ruiz Mariana

García Hernández Rogelio

Hernández Carrillo Juan Carlos

Quintero Torres Luis Leonardo

Programa 3

Analizador Sintáctico

Grupo: 2

Fecha de entrega: 13/06/2020

### Análisis del problema:

Se tienen que crear estructuras de datos para la creación de las tablas de símbolos y de tipos, las cuales tienen que ser introducidas en una pila cada una (una pila para las tablas de símbolos y una pila para las tablas de tipos)

### Implementación del programa:

El programa se implementó con tres librerías y dos archivos .c

En la librería datos.h es donde se implementan las estructuras requeridas

Estructura		Explicación
<pre>struct sym{     int dir; // dirección para la variable     int tipo; // tipo como indice a la tabla de tipos     char id[33]; // identificador     ARGS *args; // lista de argumentos     int num; // numero de argumentos     int var; // tipo de variable     SYM *next; // apuntador al siguiente simbolo };</pre>		En esta estructura se incluyen todas las partes de las tablas de símbolos El identificador La dirección El tipo Los argumentos El número de argumentos Y tipo de variable
<pre>struct sys_tab{     SYM *head;     SYM *tail;     int num; // Número de elementos en la tabla     SYMTAB *next; //apuntador a la tabla siguiente };</pre>		En esta estructura tenemos un apuntador para el inicio y el final de la tabla llevamos el número de elementos de la tabla y tenemos un apuntador que ira para la siguiente tabla de la pila
<pre>typedef struct sym_stack{     SYMTAB *top;     SYMTAB * tail; }SSTACK;</pre>		Estructura que tiene apuntadores en el tope y cola de la pila de tablas de símbolos
<pre>typedef struct tipobase{     int is_est; /* 1: es estructura 0: es tipo simple 1 no tiene tipo base*/     union{         SYMTAB *est;         int tipo;     } tipo; }TB;</pre>		Aquí definimos el tipo base de la tabla de símbolos

<pre>struct type{//tabla de tipos     int id;     char nombre[12];     TB tb;     int tam;     TYP *next; // apuntador al siguiente tipo en la tabla de tipos };</pre>	<p>En esta estructura es para los elementos de la tabla de símbolos</p> <p>El identificador</p> <p>El nombre</p> <p>El tamaño</p> <p>El tipo base</p> <p>Y el apuntador para apuntar a la siguiente tabla de tipos</p>
<pre>struct type_tab{     TYP *head;     TYP *tail;     int num;     TYPTAB *next };</pre>	<p>Estructura que contiene la variable para el conteo de elementos en la tabla</p> <p>Y el apuntador al siguiente elemento</p>
<pre>typedef struct typ_stack{     TYPTAB *top;     TYPTAB *tail; } TSTACK;</pre>	<p>Apunta a la pila de tablas de tipos, apunta al tope y a la cola de la pila.</p>

En el archivo main.c

Se verifica que las pilas y las tablas se hayan creado bien, se hacen unas pruebas introduciendo datos a la tabla de tipos, como podemos ver en las siguientes líneas del código

```
tipo1->id = 0;
```

```
strcpy(tipo1->nombre, "ent");
```

```
tipo1->tam = 8;
```

```
print_typ(tipo1);
```

```
tipo2->id = 1;
```

```
strcpy(tipo2->nombre, "real");
```

```
tipo2->tam = 8;
```

```
print_typ(tipo2);
```

```
tipo3->id = 2;
```

```
strcpy(tipo3->nombre, "dreal");

tipo3->tam = 16;

print_typ(tipo3);

printf("\n");
```

Se hace la prueba que si se han agregado a la tabla y cómo podemos observar en la imagen se han agregado los elementos a la tabla

```
Imprimiendo tabla vacia
-----
| id| nombre| tam| direccion| siguiente|
-----
|                                     La lista esta vacia|
-----

-----
| num| direccion| siguiente|
-----
| 0| 0x7fffc48c82c0| NULL|
-----

Se salio de append, la lista tiene ahora 1 elementos
Se salio de append, la lista tiene ahora 2 elementos
Se salio de append, la lista tiene ahora 3 elementos

-----
| id| nombre| tam| direccion| siguiente|
-----
| 0| ent| 8| 0x7fffc48c82f0| 0x7fffc48c8330|
-----
| 1| real| 8| 0x7fffc48c8330| 0x7fffc48c8370|
-----
| 2| dreal| 16| 0x7fffc48c8370| NULL|
-----
```

Mas adelante también se agregan elementos a las otras tablas generadas y se gregan a la pila

**Forma de ejecución:** Para ejecutar el programa ingresamos el siguiente comando en la consola de Linux:

gcc main.c tabla\_tipos.c – o estructuras

Y para ejecutar:

./estructuras

