



INSTITUTO POLITÉCNICO NACIONAL ESCUELA SUPERIOR DE CÓMPUTO

ALUMNO: MONTEALEGRE TERADA
MAXIME ISHIN
GRUPO: 3CV6

PRÁCTICA 4

Introducción

El proyecto consiste en crear un compilador que vaya del lenguaje C a ensamblador, empezando en esta práctica con el analizador léxico usando la herramienta Flex.

Flex es un generador de analizadores léxicos, una herramienta creada para generar Autómatas, en la actualidad está integrada a la herramienta Bison para en conjunto ser utilizados en la construcción de potentes compiladores.

Flex recibe un fichero de entrada o varios con extensión .l, con la descripción del escáner que se desea generar, la descripción se encuentra en forma de parejas de expresiones regulares y código en Lenguaje de Programación C, denominadas reglas, en las que definimos todas las palabras y caracteres reservados que tendrá nuestro lenguaje de programación.

Flex versión: 2.5.4.

Desarrollo

Ejemplificación del lenguaje

Aquí podemos ver un par de ejemplos que nos pueden dar una idea de cómo luce el lenguaje C

```
1. int main(void){
2.     char x;
3.     x = "Hola Mundo";
4.     printf("%s\n", x);
5. }
```

```
1. int main(void){
2.     int x, y, z;
3.     x = 1;
4.     y = 3;
5.     z = x+y;
6.     printf("%d\n", z);
7. }
```

```
1. int main(void){
2.     int veces;
3.     printf("Cuántos números quieres imprimir?");
4.     scanf("%d", &veces);
5.     for (int i = 0; i < veces; ++i){
6.         printf("%d", i);
7.     }
```

Clases léxicas

Al momento del desarrollo de la práctica, se identificaron las siguientes clases léxicas con sus respectivos elementos:

Tipo de Dato	<ul style="list-style-type: none"> • float • int • long • short • char • double
Sentencias de Control	<ul style="list-style-type: none"> • do • for • while • switch • break • continue
Operadores	<ul style="list-style-type: none"> • + • - • * • / • %
Operadores Lógicos y Relacionales	<ul style="list-style-type: none"> • > • >= • < • <= • == • != • && • • !
Asignación	<ul style="list-style-type: none"> • ++ • -- • = • *= • /= • %= • += • -= • <<= • >>= • &= • =



	<ul style="list-style-type: none">• $\wedge=$
Ellipsis	<ul style="list-style-type: none">• ...
Palabras Reservadas	<ul style="list-style-type: none">• auto• case• const• default• enum• extern• goto• register• return• signed• sizeof• static• struct• typedef• unión• unsigned• void• volatile• while
Separadores	<ul style="list-style-type: none">• ,• ;• :• ?• • ^• ~
Agrupadores	<ul style="list-style-type: none">• ()• []• {}• <>



Pruebas

Aquí se ve el proceso de compilación.

```
C:\Users\ASUS X412FA\Desktop\ESCOM\Compiladores\Prac4>flex lexico.l

C:\Users\ASUS X412FA\Desktop\ESCOM\Compiladores\Prac4>gcc -c lex.yy.c
lexico.l:112:1: warning: return type defaults to 'int' [-Wimplicit-int]
yywrap(){
^~~~~~

C:\Users\ASUS X412FA\Desktop\ESCOM\Compiladores\Prac4>gcc -c main.c
main.c: In function 'main':
main.c:2:2: warning: implicit declaration of function 'yylex' [-Wimplicit-function-
declaration]
yylex();
^~~~~~

C:\Users\ASUS X412FA\Desktop\ESCOM\Compiladores\Prac4>gcc main.o lex.yy.o

C:\Users\ASUS X412FA\Desktop\ESCOM\Compiladores\Prac4>a.exe
int
<int>
for while int "Hola Mundo"
<for><while><int><string_literal>
```

Y aquí un ejemplo de cómo identifica cada parte de un programa en C

```
C:\Users\ASUS X412FA\Desktop\ESCOM\Compiladores\Prac4>a.exe
int main (void){
<int><var><(<void>)>{
char x;
<char><var><;>
x = "Hola Mundo";
<var>=<string_literal><;>
printf("%s\n",x);
<var><(<string_literal>,<var>)<;>
}
}
```

Conclusión

Aprendí principalmente las diferencias entre cada clasificación léxica además del entender las palabras reservadas del lenguaje C.

Tuve problemas al momento de compilar todos los archivos ya que me daba errores constantes al hacerlo en Windows, esto me gastó bastante tiempo, per un compañero me recomendó agregar lo siguiente, al final en el archivo de flex:

```
yywrap(){
return(1); }
```



Así pude compilarlo cambiando la cadena “gcc main.o lex.yy.o -lfl” a “gcc main.o lex.yy.o” ya que -lfl ya está incluido ahora en el archivo.

Personalmente creo que entendí mejor cómo es que se utiliza flex y cómo se van definiendo las reglas de nuestro lenguaje de programación.

Referencias

- Manual de flex
- StackOverflow búsquedas sobre flex
- Documentación del lenguaje C de Windows