TEC | Tecnológico de Costa Rica

Instituto Tecnológico de Costa Rica

Compiladores e intérpretes

Segundo Proyecto

Scanner

Caleb Alfaro, Sebastián Quesada, Desireé Avilés

Profesor/a: Eddy Miguel Ramírez Jiménez

May 12, 2025

TEC | Tecnológico de Costa Rica

Proceso general de *scanning*

- Para la etapa de preproceso se procesa el archivo fuente de forma modular, (archivo por archivo, línea por línea, char por char). Es importante destacar que en el momento en el que se procesa un #include, se empieza a procesar inmediatamente ese archivo, y se añade al preprocesado de salida. Esto para que no solo se mantenga el orden esperado por el usuario, sino también para que la lógica de procesos para los #define, tenga el comportamiento esperado (que una vez definido, influya en todas las líneas por debajo de este).
- Los comentarios se eliminan por completo para el archivo preprocesado salida, preprocesando la línea previo al procesamiento general. Los #defines se aplican al procesar una línea, verificando si existen apariciones y remplazando por su valor, si fuera el caso.

TEC | Tecnológico de Costa Rica

Herramienta flex

Flex es una herramienta generadora de analizadores léxicos, también conocidos como escáneres. Nos facilita la creación de programas que reconocen patrones léxicos en texto, utilizando expresiones regulares. Flex toma una descripción de un analizador léxico, expresada en pares de expresiones regulares y código C (reglas), y genera un archivo fuente en C que implementa el analizador.

TEC | Tecnológico de Costa Rica

Herramienta flex

1. **Definición de reglas:** El usuario define un conjunto de reglas, donde cada regla consiste en una expresión regular y una acción en C. La expresión regular describe el patrón a buscar en el texto de entrada, y la acción en C especifica qué hacer cuando se encuentra ese patrón.

Herramienta flex

1. **Definición de reglas:** El usuario define un conjunto de reglas, donde cada regla consiste en una expresión regular y una acción en C. La expresión regular describe el patrón a buscar en el texto de entrada, y la acción en C especifica qué hacer cuando se encuentra ese patrón.
2. **Generación de código C:** Flex toma estas reglas y genera un archivo fuente en C que contiene la implementación del analizador léxico. Este archivo contiene la lógica para encontrar y procesar los patrones definidos en las reglas.

TEC | Tecnológico de Costa Rica

Herramienta flex

1. **Definición de reglas:** El usuario define un conjunto de reglas, donde cada regla consiste en una expresión regular y una acción en C. La expresión regular describe el patrón a buscar en el texto de entrada, y la acción en C especifica qué hacer cuando se encuentra ese patrón.
2. **Generación de código C:** Flex toma estas reglas y genera un archivo fuente en C que contiene la implementación del analizador léxico. Este archivo contiene la lógica para encontrar y procesar los patrones definidos en las reglas.
3. **Compilación y uso:** El archivo fuente en C generado por Flex se compila con un compilador de C y se utiliza para crear un programa que realiza la tarea de analizar léxicamente el texto de entrada.

TEC | Tecnológico de Costa Rica

Programa después del preproceso

```
1  int holaAndamosTesting(long long xd);
2  long long largo(long long grueso);
3  int mokeyplus(int a, int b) {
4  a+=b;
5  return a + b;
6  }#errortoken nosirve
7  int main() {
8  double numPi = 3.1416;
9  double dosPi = 3.1416 * 2;
10 long long mesi= "testmessi";
11 char* mesi2 = "testmessi2";
12 return 0;
13 }
```

TEC | Tecnológico de Costa Rica

Histograma de las cantidades de tokens

Categoría Léxica	Cantidad
Palabras Reservadas	18
Operadores y símbolos	31
Literales e identificadores	23
Errores y fin de archivo	2

TEC | Tecnológico de Costa Rica

Cantidades de palabras reservadas

Palabra	Cantidad	Palabra	Cantidad
break	0	case	0
char	1	const	0
continue	0	default	0
do	0	double	2
else	0	enum	0
extern	0	float	0
for	0	goto	0
if	0	int	5
long	8	register	0
return	2	short	0
signed	0	sizeof	0
static	0	struct	0

TEC | Tecnológico de Costa Rica

Cantidades de palabras reservadas

Palabra	Cantidad	Palabra	Cantidad
switch	0	typedef	0
union	0	unsigned	0
void	0	volatile	0
while	0	long long	0
long double	0	bool	0
true	0	false	0
restrict	0	inline	0
complex	0	imaginary	0
thread_local	0	atomic	0
noreturn	0		

TEC | Tecnológico de Costa Rica

Cantidades de los operadores y símbolos

Palabra	Cantidad	Palabra	Cantidad
plus	2	minus	0
mult	2	div	0
assign	5	eq	0
neq	0	lt	0
gt	0	leq	0
geq	0	and	0
or	0	not	0
bitand	0	bitor	0
xor	0	complement	0

TEC | Tecnológico de Costa Rica

Cantidades de los operadores y símbolos

Palabra	Cantidad	Palabra	Cantidad
lshift	0	rshift	0
increment	0	decrement	0
dot	0	arrow	0
lparen	4	rparen	4
lbrace	2	rbrace	2
lbracket	0	rbracket	0
semicolon	9	colon	0
comma	1	ellipsis	0

TEC | Tecnológico de Costa Rica

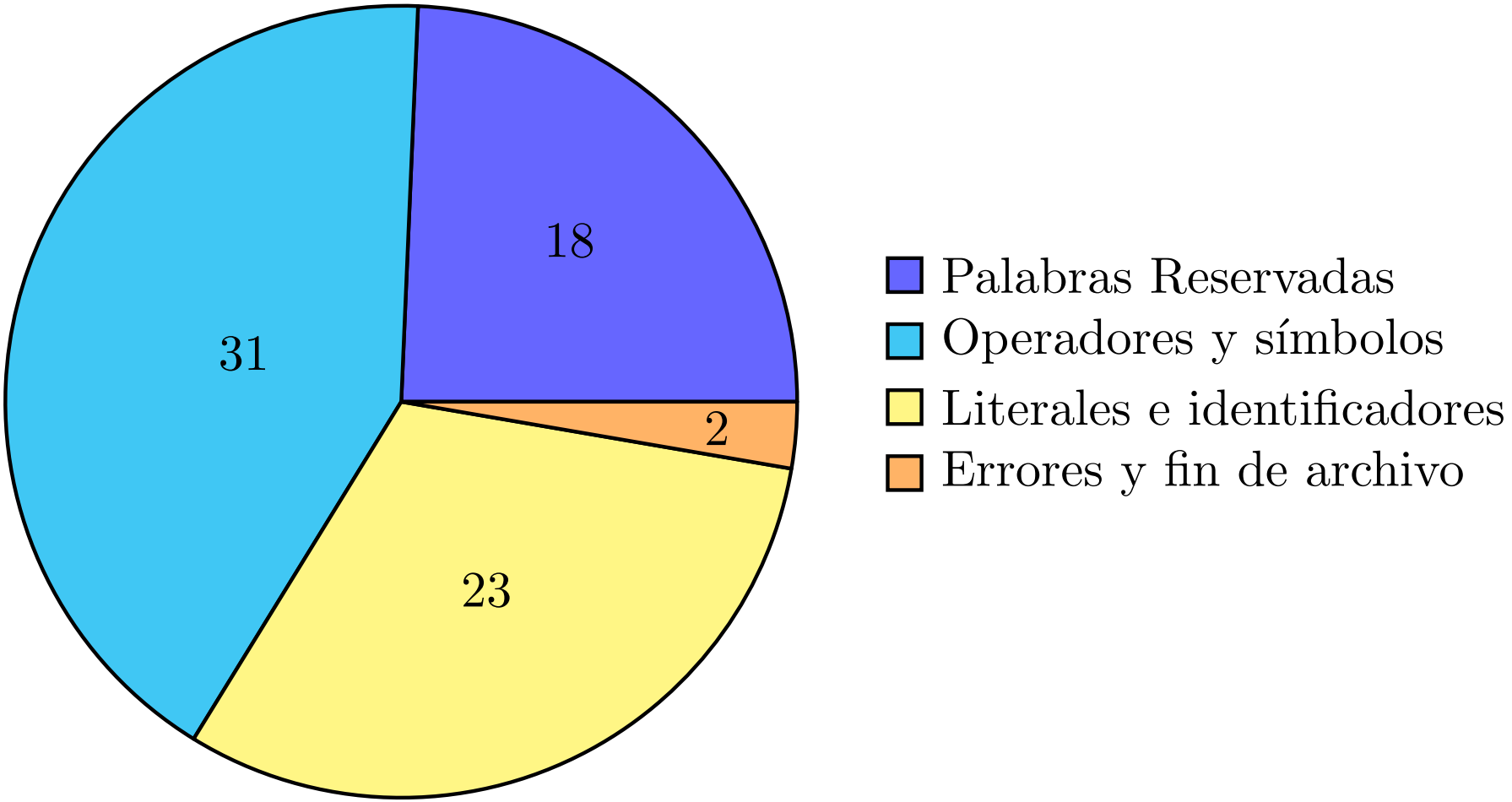
Cantidades de literales e identificadores, errores y final de archivo

Palabra	Cantidad
identifier	17
intliteral	2
floatliteral	2
stringliteral	2
error	1
eof	1
char literal	0
bool literal	0

TEC | Tecnológico de Costa Rica

Grafico de pastel

Cantidad de cada categoría léxica



TEC | Tecnológico de Costa Rica

Bibliografía y fuentes consultadas

- Universidad de Zaragoza. (2003-2004). *Introducción a Flex y Bison.* Departamento de Informática e Ingeniería de Sistemas. https://webdiis.unizar.es/asignaturas/LGA/material_2003_2004/Intro_Flex_Bison.pdf
- Free Software Foundation. (s.f.). *Flex: The Fast Lexical Analyzer.* Massachusetts Institute of Technology. (Trabajo original en inglés, traducido al español). https://web-mit-edu.translate.goog/gnu/doc/html/flex_1.html?_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es&_x_tr_pto=sge

TEC | Tecnológico de Costa Rica