

Práctica 2 - Compiladores

Hecho por: *Elvis André Cruces Gómez y Yoshiro Milton Miranda Valdivia*

Ejercicio 1

1. Redacte el siguiente código en dos archivos distintos, luego compile cada archivo y explique a qué se debe el tipo de error del segundo código. **(2 puntos)**.

```
int main(){
    char* c = "abcdef";
    int m = 11148;
    return 0;
}
```

```
int main(){
    char* c = "abcdef";
    int 4m = 11148;
    return 0;
}
```

Solución:

Compilador (4)				Recursos	Registro de Compilación	Depuración	Resultados	Cerrar
Línea	Col.	Archivo	Mensaje					
3	6	D:\La Salle\Cursos André\VI Semestre\Compiladores\Practica ...	[Error] invalid suffix "m" on integer constant					
		D:\La Salle\Cursos André\VI Semestre\Compiladore...	In function 'int main()':					
2	12	D:\La Salle\Cursos André\VI Semestre\Compiladores\Practica ...	[Warning] deprecated conversion from string constant to 'char*' [-Wwrite-strings]					
3	6	D:\La Salle\Cursos André\VI Semestre\Compiladores\Practica ...	[Error] expected unqualified-id before numeric constant					

El principal y único error que pudimos encontrar en el segundo código, es que no se puede iniciar el nombre de una variable con un valor numerico, ya que el sistema lo detecta como un número más. En resumen, el error se encuentra en la línea de código 3 del segundo código (`int 4m = 11148;`).

Ejercicio 2

2. Explique cuál es la función del Scanner o Analizador Léxico. (2 puntos)

Solución:

El analizador léxico es la primera fase de un compilador. Su principal función consiste en leer los caracteres de entrada y elaborar como salida una secuencia de componentes léxicos que utiliza el analizador sintáctico para hacer el análisis. Esta interacción, suele aplicarse convirtiendo al analizador léxico en una subrutina o corrutina del analizador sintáctico. Y bueno también tenemos funciones secundarias como:

- Eliminar los comentarios del programa.
- Eliminar espacios en blanco, tabuladores, etc, y en general, todo aquello que carezca de significado según la sintaxis del lenguaje.
- Reconocer los identificadores de usuario, números, palabras reservadas del lenguaje y tratarlos correctamente con respecto a la tabla de símbolos (solo en los casos que debe de tratar con la tabla de símbolos).
- Llevar la cuenta del número de línea por la que va leyendo, por si se produce algún error, dar información sobre donde se ha producido.
- Avisar de errores léxicos. Por ejemplo, si @ no pertenece al lenguaje, avisar de un error.
- Puede hacer funciones de preprocesador.

Ejercicio 3

3. Dado el siguiente código, en una tabla escriba todos los Tokens que un Scanner encontraría, detalle la clase y el lexema. Usted tiene la libertad de escoger el nombre de las clases. (4 puntos).

```
int main(){
    char* c = "abcdef";
    int m = 11148;
    int x = m/8;
    int y = m/4;
    int z = m/2;
    return 0;
}
```

Solución:

CLASS	LEXEME
type_int	int
id	main
par_((
par_))
key_{	{
type_char	char
ope_mul	*
id	c
ope_equal	=
com_“	“
text	abcdef
com_”	”
dot_com	;
type_int	int
id	m
ope_equal	=
num	11148
dot_com	;
type_int	int
id	x
ope_equal	=

id	m
ope_div	/
num	8
dot_com	;
type_int	int
id	y
ope_equal	=
id	m
ope_div	/
num	4
dot_com	;
type_int	int
id	z
ope_equal	=
id	m
ope_div	/
num	2
dot_com	;
ope_ret	return
num	0
dot_com	;
key_}	}

Ejercicio 4

4. Redacte el siguiente código en dos archivos distintos, luego compile cada archivo y explique a qué se debe el tipo de error del segundo código. (2 puntos).

```
int main(){  
    int m = 11148;  
    int x = m + 100;  
    return 0;  
}
```

```
int main(){  
    int m = 11148;  
    int x = m + ;  
    return 0;  
}
```

Solución:

Compilador (2) Recursos Registro de Compilación Depuración Resultados Cerrar			
Línea	Col.	Archivo	Mensaje
		D:\La Salle\Cursos André\VI Semestre\Compiladores\...	In function 'int main()':
3	14	D:\La Salle\Cursos André\VI Semestre\Compiladores\Pra...	[Error] expected primary-expression before ';' token

El principal error que encontramos e identificamos en el segundo código del ejercicio 4, es que en la línea de código 3, la variable "x" es igual a "m +", no se le agrega ni otra variable ni un número. El error que nos muestra el compilador usado es que se esperaba un valor o expresión primario antes del ";". Por lo tanto, no se ejecuta ninguna operación por falta de información o datos.

Ejercicio 5

5. Explique cuál es la función del Parser o Analizador Sintáctico. (2 puntos)

Solución:

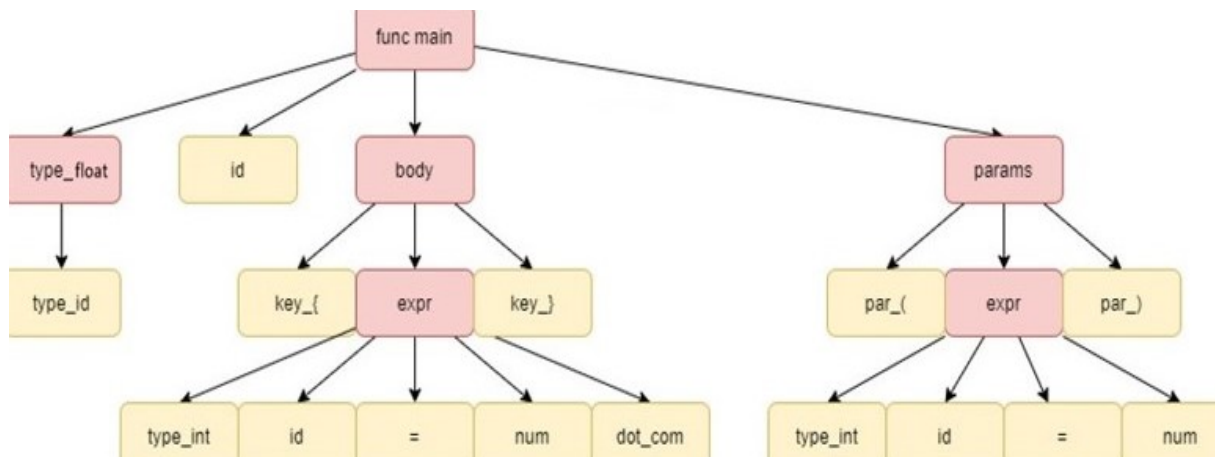
Un analizador sintáctico o parser es un programa que normalmente es parte de un compilador. El compilador se asegura de que el código se traduce correctamente a un lenguaje ejecutable. La tarea del analizador es, en este caso, la descomposición y transformación de las entradas en un formato utilizable para su posterior procesamiento. Se analiza una cadena de instrucciones en un lenguaje de programación y luego se descompone en sus componentes individuales.

Ejercicio 6

6. Genere el árbol sintáctico para el siguiente código. Puede tomar como ejemplo el mostrado en clases. (4 puntos).

```
float main(int a = 5){  
    int x = 100;  
}
```

Solución:



Ejercicio 7

7. Redacte el siguiente código en dos archivos distintos, luego compile cada archivo y explique a qué se debe el tipo de error del segundo código. **(2 puntos)**.

```
int main(){  
    int m = 8;  
    int y = 3;  
    int x = m + y;  
    return 0;  
}
```

```
int main(){  
    int m = "ceee";  
    int x = m + y;  
    return 0;  
}
```

Solución:

Compilador (3) Recursos Registro de Compilación Depuración Resultados Cerrar			
Línea	Col.	Archivo	Mensaje
		D:\La Salle\Cursos André\VI Semestre\Compiladores\...	In function 'int main()':
2	10	D:\La Salle\Cursos André\VI Semestre\Compiladores\Pra...	[Error] invalid conversion from 'const char*' to 'int' [-fpermissive]
3	14	D:\La Salle\Cursos André\VI Semestre\Compiladores\Pra...	[Error] 'y' was not declared in this scope

En este ejercicio pudimos encontrar 2 errores en el segundo código. El primero se encuentra en la línea de código 2, donde nos muestra que hubo una conversión inválida de 'char*' a 'int', ya que el tipo de valor que tiene la variable m es un entero (int), mientras que el valor que se le asigno es un grupo de letras (char*). El segundo error está en la línea de código 3, que nos muestra el error de que la variable *z*", que se usa en la operación, no ha sido declarada anteriormente, en otras palabras, no existe.

En resumen, el primer error se debe a un mal asignamiento de valor respecto al tipo de valor que se requiere, nos se puede asignar un 'char*' a un 'int' y el segundo error se debe a que no se declaró una variable que se iba a usar en esa operación.

Ejercicio 8

8. Explique cuál es la función del Analizador Semántico. (2 puntos)

Solución:

El analizador semántico utiliza el árbol sintáctico y la información en la tabla de símbolos para comprobar la consistencia semántica del programa fuente con la definición del lenguaje. También recopila información sobre el tipo y la guarda, ya sea en el árbol sintáctico o en la tabla de símbolos, para usarla más tarde durante la generación de código intermedio.

Una parte importante del análisis semántico es la comprobación (verificación) de tipos, en donde el compilador verifica que cada operador tenga operando que coincidan. Por ejemplo, muchas definiciones de lenguajes de programación requieren que el índice de un arreglo sea entero; el compilador debe reportar un error si se utiliza un número de punto flotante para indexar el arreglo.