

Actividad 03. Analizador léxico

Implementar, utilizando Lex, un analizador léxico que, a partir de un fichero de entrada, genere un fichero de salida según la siguiente descripción.

Teniendo en cuenta la siguiente tabla, se sustituirán los elementos que se describen en la primera columna por el texto que aparece en la segunda.

Palabras reservadas: int, float, return, for, if, else	Aparecerá la palabra correspondiente escrita en mayúsculas
Identificadores: palabras que pueden tener letras, dígitos y <code>_</code> , pero no pueden comenzar por un dígito	ID
Números enteros	ENTERO
Números reales con notación clásica o científica, admitiendo también números que carezcan de la parte entera o decimal	REAL
Cadenas de caracteres (texto encerrado entre dobles comillas). Para simplificar el problema suponemos que una cadena no puede ocupar varias líneas	TEXTO

Se reconocerán los números sin signo (excepto en el caso del exponente de la notación científica).

Además, el analizador debe cumplir los siguientes requisitos:

1. Se deben eliminar los comentarios y las líneas en blanco. Como en C/C++, hay dos tipos de comentarios:
 - Los que comienzan con `//` y acaban con la línea
 - Los que están encerrados entre `/*` y `*/`, que pueden ocupar varias líneas
2. Los demás caracteres (+, *, ;, ...) deben aparecer igual en el fichero de salida.
3. El analizador tendrá dos parámetros: el primero será el nombre del fichero de entrada y el segundo el del fichero de salida.
4. Se debe definir una estructura de datos que almacene los nombres de todos los identificadores y el número de la línea en la que aparece por primera vez en el texto (fichero de entrada).
 - a. La definición de la estructura de datos y de las operaciones básicas para trabajar con ella se llevará a cabo en una biblioteca **independiente** del analizador léxico.
 - b. Aunque el mismo identificador aparezca varias veces en el fichero de entrada, solo se almacena una vez en la estructura.
 - c. La estructura de datos puede ser estática o dinámica, y podemos suponer que, como máximo, habrá que almacenar 100 identificadores.
5. En el fichero de salida, al final, debe aparecer el contenido de la estructura de datos, es decir, los nombres de todos los identificadores y el número de línea en la que aparece por primera vez. También debe aparecer el número total de líneas del fichero de entrada.

Ejemplo de funcionamiento:

Si el contenido del fichero de entrada es el siguiente:

```
/* Esto es un bucle */  
  
for (i=0; i < 10; i++)    // segundo comentario  
  
    x = x * 2.5 + i;  
return x;
```

En el fichero de salida debe aparecer:

```
FOR (ID=ENTERO; ID < ENTERO; ID++)  
ID = ID * REAL + ID;  
RETURN ID;  
  
Lista de identificadores:  
  
i      3  
x      5  
  
Número total de líneas del fichero de entrada: 5
```

La entrega de la actividad debe estar formada por:

1. El analizador léxico (.l)
2. La biblioteca con la definición de la estructura de datos que almacena los identificadores
3. El fichero makefile que, además de generar el analizador, debe ejecutarlo usando como fichero de entrada **entrada.txt** y de salida **salida.txt**.

Esta actividad se puede realizar en pareja. El nombre de las dos personas deberá aparecer en un comentario dentro del fichero makefile. Ambos miembros de la pareja deberán entregar la tarea en el aula virtual.

Esta actividad se debe entregar (a través del aula virtual) antes del **jueves, 16 de marzo de 2023, a las 23:55 horas**.