

Sesión 02. Make y Lex (2ª parte)

Objetivos

- Utilización de make
- Ampliación de Lex

Recursos que se utilizarán en esta sesión (disponibles en el aula virtual)

- Guion de la sesión (*este documento*)
- Make
 - o Presentación en PDF
 - o Vídeo con la explicación
- Lex (2ª parte)
 - o Presentación en PDF
 - o Vídeo con la explicación
- Ficheros de ejemplo: makefile, entrada.txt, simple.l, contador.l

Trabajo durante la sesión

Ejercicio 01. Prueba de ficheros de ejemplos

- Probar el funcionamiento de **make** para generar el analizador léxico **simple**.
- Ejecutar el analizador **simple** redireccionando la entrada estándar al fichero **entrada.txt** y redireccionando la salida estándar al fichero de salida **salida.txt**.
- Analizar las diferencias entre ambos ficheros **entrada.txt** y **salida.txt**.

Ejercicio 02. Definición de un fichero makefile

Modificar el fichero **makefile** para que permita generar el ejecutable del analizador léxico **contador** a partir del fichero **contador.l**.

Se puede intentar, además, que, **make** ejecute **contador** redirigiendo la entrada estándar del fichero **entrada.txt** y la salida al fichero **salida.txt**.

A partir de ahora, se debe definir siempre un fichero makefile para hacer todas las actividades y ejercicios que se planteen.

(Se puede utilizar como fichero **contador.l** el resultado de la sesión anterior.)

Ejercicio 03. Modificación de simple.l para contar nuevos *tokens*

Modificar el programa **contador.l** para que cuente el número de apariciones de:

- Palabras reservadas. Consideraremos solo: int, float, return, for, if, else
- Identificadores (palabras que pueden tener letras, dígitos y `_`, pero no pueden comenzar por un dígito)
- Números enteros

- Números reales con notación clásica (5.7) y científica (5.7e+8), admitiendo también números que no tengan parte entera (.7) o parte decimal (5.)

Por ahora no se van a tener en cuenta la posibilidad de que aparezcan números con signo, excepto en el exponente de la notación científica.

- Cadenas de caracteres (texto encerrado entre dobles comillas)
- Líneas en blanco
- Líneas con información

El programa solo mostrará los resultados calculados, es decir, no debe escribir ningún carácter extraño en la pantalla.

Probar el funcionamiento del programa con el ejemplo **entrada.txt**. El resultado debería ser similar a éste:

Número de palabras reservadas: 7
Número de identificadores: 24
Número de enteros: 3
Número de reales: 4
Número de cadenas de caracteres: 4
Número de líneas en blanco: 7
Número de líneas con información: 16

Es fundamental ir construyendo un fichero de pruebas propio que incluya ejemplos suficientes de todos los *tokens* que se quieren detectar, variantes, contraejemplos, etc.