

MEMORIA PROYECTO LEX

Sofía Fernández Moreno
Modelos de computación
3º C
Curso 2015-2016



ugr

Universidad
de **Granada**

1. Descripción del problema planteado.

El problema que he planteado es un conversor de texto plano a HTML, el cual le he asignado un diseño con un fichero CSS externo.

El proyecto se encarga de detectar cualquier expresión y la introduce en un fichero HTML.

La implementación del mismo ha sido crear dos funciones las cuales se encargan de realizar la apertura y el cierre de nuestro “body” (cuerpo del fichero HTML). Dentro de estas dos funciones irá nuestra implementación Lex.

Las dos funciones mencionadas antes, serían para la apertura del “body” de un fichero HTML, cuyo formato sería (la parte de azul representaría nuestra apertura.):

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<HTML>
  <HEAD>
    <META http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <LINK href="colores.css" rel="stylesheet" type="text/css">
  </HEAD>
  <BODY><SPAN CLASS='numerodelinea'>1</SPAN><br /><SPAN CLASS='include'>#line</SPAN>&nbsp;<SPAN CLASS='numero'>3</SPAN>&nbsp;<SPAN CLASS='cadena'>"lex.yy.c"</SPAN><SPAN
```

Ilustración 1 Ejemplo del código fuente de un fichero HTML

Ahora para el cierre de nuestro “body”, su formato sería:

</BODY>

</HTML>

En la sección de reglas, he utilizado los fprintf para que dentro del fichero CSS vaya recogiendo el diseño que quiero asignarle a cada regla que he declarado. Hay distintos tipos para cada sección, incluso puede ir asignadas con un código, ejemplos:

Para los tabuladores--> 	

Para los espacios en blanco-->

Los demás tipos los podemos encontrar en la siguiente URL:

<http://www.rapidtables.com/web/html/html-codes.htm>

Para hacer uso de nuestro fichero CSS externo hacemos mención de > y entre esas comillas mencionaremos cualquiera de las funciones correspondientes de nuestro archivo.

2. Compilación práctica

Desde la carpeta de nuestro programa introducimos en la terminal:

```
lex traductor.l //este generará un archivo lex.yy.c
gcc -o traductor lex.yy.c -ll //Este generará nuestro programa
```

3. Uso del programa

Al generarse el ejecutable traductor, debemos meterle por salida estándar el archivo que queremos traducir a HTML.

Ejemplo:

```
./salida defs.cpp
```

Aquí se generará un archivo “fichero.html” el cual abriremos en nuestro navegador obteniendo la siguiente salida:

```
1  #include "cabeceras.h"
2
3  void indent (string text,int nindent){
4      if (one_line_block){
5          one_line_block = false;
6          indentation_lvpop ();
7      }
8
9      for (int i=0; i<nindent; ++i){
10         cout << tab;
11     }
12     cout << text;
13 }
14
15 int look_for (string text, char c){
16     bool encontrado = false;
17     int i;
18
19     for (i=0; i<textlength () && !encontrado; ++i){
20         if (text[i] == c)
21             encontrado = true;
22     }
23
24     return i;
25 }
26
27 int Stack::top (){
28     if (!pilaempty ())
29         return pilaback ();
30     else
31         return 0;
32 }
33
34 void Stack::pop (){
35     if (!pilaempty ())
36         pilapop_back ();
37 }
38
39 void Stack::push (int n){
40     pilapush_back (n);
```

Ilustración 2 Fichero HTML

Bibliografía

<http://www.w3schools.com/html/>

<http://flex.sourceforge.net/manual/EOF.html#EOF>

<http://www.rapidtables.com/web/html/html-codes.htm>

<http://www.w3schools.com/css/>