



UNIVERSIDAD DE GRANADA

Modelos de Computación

E.T.S. de Ingenierías Informática y de Telecomunicación

Memoria práctica 2 Flex

Problema abordado

Muchos alumnos y personal de la UGR utilizan los servicios del comedor universitario con frecuencia. Sería útil una aplicación para consultar los menús disponibles con la información actualizada, pero no hay ninguna API pública para realizar esta función.

Solución realizada

Usando Flex podemos conseguir esta información gracias a las expresiones regulares. Descargando la página web de los comedores universitarios (scu.ugr.es) y obteniendo solo las partes que nos interese con expresiones regulares.

Gracias a las acciones asociadas podremos eliminar la parte html que no queremos mostrar y escoger solo los menús de los días que queremos consultar o los que hay en un lugar específico.

Se ha incluido en la entrega de la práctica un archivo html de ejemplo y una opción en el programa para usar el fichero en vez de la pagina web de los comedores por si en el momento de corrección el servicio de comedores está cerrado.

Como solo vamos a analizar un fichero de entrada especificamos la opción noyywrap.

Sección de declaraciones

Tenemos dos tipos de declaraciones en dos bloques.

El primer bloque (entre los símbolos “%{“ y “%}”) sirve para añadir el código tal cual al fichero c++ generado por flex. Aquí incluimos las ficheros de cabecera que vamos a necesitar y variables globales, cuya utilidad está descrita en la siguiente captura.

```

1  ▼ %option noyywrap
2
3  ▼ %{
4      #include <iostream>
5      #include <fstream>
6      #include <iomanip>
7      #include <sstream>
8      #include <ctime> // Librería para obtener el tiempo actual del sistema y
9      >> >> >> // usarlo para consultar automáticamente el día actual
10
11     #include <stdio.h>
12     #include <string.h>
13     #include <stdlib.h>
14     #include <string>
15     #include <curl/curl.h> // Librería para descargar la página web de comedores en html
16
17     using namespace std;
18
19     // Variables globales
20     string encontrado;>
21     yyFlexLexer *flujo; // Objeto de la clase FlexLexer que sirve para recibir el texto y la longitud del token >> >>
22     >> >> >> >> >> reconocido más recientemente y para obtener el número de línea de entrada actual
23
24     string n_day;> // String donde guardaremos el día del mes (de 1 a 31) para comparar con el día de los menús
25
26     bool pts = false;> >> // Booleano que indica a flex si mostrar los menús del pts o los menús generales.
27     bool escribir = true;
28     bool day = true;> >> // Booleano que indica a flex si mostrar solo el menú del día guardado en n_day o todos >>
29     >> >> >> >> >> los menús
30     bool day_found = false; // Booleano que cambia su valor a true cuando el día encontrado en una fecha coincide con
31     >> >> >> >> >> el guardado en n_day
32
33     %}

```

En el segundo bloque definimos alias para expresiones regulares que nos serán útiles en la sección de reglas.

```

32  n_dia> [0-9]|1[0-9]|2[0-9]|3[0-1]
33  día>> (LUNES|MARTES|MIÉRCOLES|JUEVES|VIERNES|SÁBADO|DOMINGO)
34  mes>> (ENERO|FEBRERO|MARZO|MAYO|ABRIL|JUNIO|JULIO|AGOSTO|SEPTIEMBRE|OCTUBRE|NOVIEMBRE|DICIEMBRE)
35  fecha> {día}," "{n_dia}" "DE" "{mes}" "DE" "{entero}"
36  menu> Menú" "[1-3]
37  ele>> (Entrante|Primero|Segundo|Acompañamiento|Postre|Bebida)" </td><td class="leftalign"><strong>."+</
38  pts>>> "Menú Comedor (PTS)</a></h1><div class="dokuwiki_backToTop"><a href="#" class="action top"
39  >> >> >> >> >> accesskey="x" >Subir"

```

n_dia acepta un número del 1-31

día acepta un día de la semana

mes acepta un mes

fecha acepta una fecha con el formato en el que están en la página web del scu

ele acepta un elemento del menu, con el html que le rodea, el tipo (Primero, segundo, postre...) y el nombre de la comida.

pts acepta la cadena en html que aparece en el código html previamente a los menús del pts

Sección de reglas

En esta sección asociamos a expresiones regulares acciones en c++ que se ejecutarán cuando se encuentren en el fichero de entrada una entrada aceptada por la expresión regular.

```
40  ▼ %%  
41  
42  ▼ {pts}» {  
43  » » » if (pts)  
44  » » »     escribir = true;  
45  » » » else  
46  » » »     return 0;  
47  » » » }  
48  ▼ {fecha}» {  
49  » » » encontrado = flujo->YYText();  
50  » » » if (escribir)  
51  » » » {  
52  » » »     if (day)  
53  » » »     {  
54  » » »         if (encontrado.substr(encontrado.find(", ") + 3, 2) == n_day)  
55  » » »         {  
56  » » »             day_found = true;  
57  » » »             cout << "\n" << encontrado << endl;  
58  » » »         }  
59  » » »         else  
60  » » »             day_found = false;  
61  » » »         }  
62  » » »         else  
63  » » »             cout << "\n" << encontrado << endl;  
64  » » »     }  
65  » » » }  
66  ▼ {menu}» {  
67  » » » if (escribir && (day && day_found || !day))  
68  » » »     cout << "\n" << flujo->YYText() << endl;  
69  » » » }  
70  ▼ {ele}» {  
71  » » » encontrado = flujo->YYText();  
72  » » » if (escribir && (day && day_found || !day))  
73  » » » {  
74  » » »     cout << setw(20) << left << encontrado.substr(0, encontrado.find(" "));  
75  » » »     cout << setw(20) << left << encontrado.substr(encontrado.find("<strong>") + 8,  
76  » » »     encontrado.find("</strong>") - (encontrado.find("<strong>") + 8)) << endl;  
77  » » » }  
78  » » » }  
79  (.\n)» {}  
80  ▼ %%
```

La regla asociada a la expresión regular del alias {pts} se activa cuando llegamos a la parte del fichero en la que se encuentran los menús del pts (que está después de los menús generales).

Si hemos pasado la opción al programa de mostrar los menús del pts pondrá escribir a true para permitir que se muestren y si no finalizaremos el programa (por que ya se habrán mostrado los menús generales).

La regla asociada a la fecha se activa cuando llegamos a la parte del fichero en la que se encuentra una fecha asociada a un menú.

Si la variable “escribir” tiene valor “true”, porque esta fecha está en la parte de menús generales o la de menú pts y esa es la parte que queremos mostrar continuamos.

Si la variable “day” tiene valor “true” comprobaremos que el día de la fecha coincide con “n_day” y entonces mostraremos la fecha y cambiaremos el valor de “day_found” a “true, si no coincide solo cambiaremos el valor de “day_found” a false. En cambio si la variable “day” tiene valor “false” mostraremos la fecha sin más comprobaciones.

La regla asociada a {menu} mostrará el número de menú si hay que escribir los menús de esta zona y si este pertenece a una fecha deseada.

La regla asociada a {ele} mostrará el elemento del menu si hay que escribir los menús de esta zona y si este pertenece a una fecha deseada, eliminando la parte html aceptada con la expresión regular y estableciendo un formato adecuado con la misma anchura en la salida independientemente del nombre del elemento encontrado.

Con la regla asociada a la expresión regular (.\n) no realizamos ninguna acción para todos los caracteres y saltos de líneas que no sean aceptados por ninguna de las reglas anteriores.

Sección de procedimientos

Finalmente en la sección de procedimientos escribimos todo el código c++ necesario.

En el main procesaremos los distintos argumentos pasados al programa para escoger varias opciones, obtendremos la hora del sistema por si necesitamos consultar el día actual, descargaremos la página web en html de “<https://scu.ugr.es/>” y se la pasaremos al analizador de Flex.

Ejemplo de ejecución

Finalmente la ejecución se realizaría de la siguiente forma:

Modo de empleo: scu [OPCION]... [DIA]

La opción por defecto es mostrar el menú general del día actual

-a, --all	muestra todos los días disponibles
-h, --help	muestra esta ayuda y finaliza
-p, --pts	muestra los menús del comedor del pts
-f <file>	lee la información desde el fichero proporcionado

El argumento DIA es un número entero entre 1 y 31, ambos incluidos.

Si se proporciona se mostrarán solo los menús de este día si existe alguno.

A continuación se muestra en capturas de pantalla varios ejemplos de ejecución con distintas opciones.

[dipzza@dipzza-pc 2]\$./scu -a

Menú Comedor

SÁBADO, 14 DE DICIEMBRE DE 2019

Menú 1

Primero	Pizza
Segundo	Jamón asado en salsa de strogonoff
Acompañamiento	Menestra
Postre	Natillas caseras
Bebida	Vino tinto

LUNES, 16 DE DICIEMBRE DE 2019

Menú 1

Primero	Pizza
Acompañamiento	Ensalada
Segundo	Lomo braseado
Acompañamiento	Berenjenas fritas (VG)
Postre	Naranja I
Bebida	Vino tinto

MARTES, 17 DE DICIEMBRE DE 2019

Menú 1

Primero	Cocido
Acompañamiento	Pincho de tortilla de patatas y atún sobre ajo blanco
Segundo	Merluza a la panadera
Postre	Piña
Bebida	Vino tinto

Menú 2

Primero	Cocido vegetariano (OV)
Acompañamiento	Pincho de tortilla española con ajo blanco
Segundo	Rollitos de primavera
Acompañamiento	Ensalada de col
Postre	Piña
Bebida	Vino tinto

MIÉRCOLES, 18 DE DICIEMBRE DE 2019

Menú 1

Primero	Sopa de mi pueblo
Segundo	Plato alpujarreño
Postre	Manzana
Bebida	Vino tinto

Menú 2

Primero	Sopa minestrone
Segundo	Queso rebozado en crocante
Acompañamiento	Patatas a lo pobre y huevo frito
Postre	Manzana
Bebida	Vino tinto

JUEVES, 19 DE DICIEMBRE DE 2019

Menú 1

Primero	Macarrones a la boloñesa
Acompañamiento	Menestra de verduras
Postre	Mandarinas
Bebida	Vino tinto

```
[dipzza@dipzza-pc 2]$ ./scu 18
Menú Comedor

MIÉRCOLES, 18 DE DICIEMBRE DE 2019

Menú 1
Primero      Sopa de mi pueblo
Segundo      Plato alpujarreño
Postre       Manzana
Bebida       Vino tinto

Menú 2
Primero      Sopa minestrone
Segundo      Queso rebozado en crocante
Acompañamiento Patatas a lo pobre y huevo frito
Postre       Manzana
Bebida       Vino tinto
```

```
[dipzza@dipzza-pc 2]$ ./scu --pts 18
Menú Comedor (PTS)

MIÉRCOLES, 18 DE DICIEMBRE DE 2019

Menú 1
Primero      Sopa de mi pueblo
Segundo      Plato alpujarreño
Postre       Manzana
Bebida       Vino tinto

Menú 2
Primero      Sopa minestrone
Segundo      Queso rebozado en crocante
Acompañamiento Patatas a lo pobre y Huevo frito
Postre       Manzana
```

```
[javierb@diplaptop 2]$ ./scu -h
Modo de empleo: scu [OPCION]... [DIA]
Consulta el Servicio de Comedores Universitarios de la UGR

La opción por defecto es mostrar el menú general del día actual

-a, --all      muestra todos los días disponibles
-h, --help     muestra esta ayuda y finaliza
-p, --pts     muestra los menús del comedor del pts
-f <file>     lee la información desde el fichero proporcionado

El argumento DIA es un número entero entre 1 y 31, ambos incluidos.
Si se proporciona se mostrarán solo los menús de este día si existe alguno.
```