

MODELOS DE COMPUTACIÓN PRÁCTICA 2

LEX COMO LOCALIZADOR DE EXPRESIONES REGULARES CON ACCIONES ASOCIADAS



UNIVERSIDAD
DE GRANADA

Realizado por: Francisco Beltrán Sánchez y Miguel Ángel Posadas Arráez

Índice

- 1.Descripción del problema
- 2.Explicación de como se ha abordado el tema
- 3.Capturas de pantalla del funcionamiento

1. Descripción del problema

El problema que hemos tratado en esta práctica es el de validación de un número de teléfono.

Hemos considerado que los números válidos serán aquellos que empiecen por 6 u 8 y que contengan una cantidad exacta de 9 caracteres (no hemos considerado el prefijo de la nacionalidad del número).

En definitiva los números de telefonía móvil válidos en España.

2. Explicación de como se ha abordado el tema

Para abordar el problema hemos utilizado lex que es una herramienta para generar analizadores léxicos junto al programa yacc para generar un análisis sintáctico.

En la primera sección tenemos las declaraciones:

```
/* Declaraciones */
%{
#include <stdio.h>

int correcto, incorrecto;
void resultados(int a, int b);
%}

primer_digito    [6|8]
digitos          [0-9]
```

Primeramente incluimos la librería stdio.h (la librería estándar de entrada y salida de C).

Después creamos dos contadores que son los que llevarán la cantidad de números de teléfono son correctos y cuáles incorrectos.

Luego declaramos la cabecera de la función resultados que es la que se encarga de imprimir por pantalla los resultados, en la sección de procedimientos se verá su implementación.

Finalmente definimos primer_digito que contiene los números que aceptamos como válidos para el primer dígito del número de teléfono y el intervalo digitos que contiene los dígitos válidos después del primero (aceptamos todos los números entre 0 y 9).

```

/* Reglas */
%%

{primer_digito}{digitos}{8} {printf("%s es correcto\n",yytext);correcto++;}
{digitos}+                  {printf("%s es incorrecto\n",yytext);incorrecto++;}
.+                          {printf("%s no es un numero\n",yytext);incorrecto++;}
%%

```

Ahora viene la sección de reglas, aquí definimos qué números son correctos y cuáles no.

En la primera línea vemos que un número es correcto si está compuesto por un dígito perteneciente a primer_digito (previamente definido) y por una secuencia de dígitos pertenecientes al intervalo del que hablamos antes, concretamente una secuencia de 8 dígitos pertenecientes a dicho intervalo (8 + el primer dígito)

En la segunda línea declaramos que si es una secuencia de números que no necesariamente comience por lo que consideramos como primer dígito correcto y de una longitud no definida, no se considerará como correcto.

Finalmente en la tercera declaramos que cualquier secuencia que contenga algo que no sea un número no será correcto y además se mostrará por pantalla que efectivamente lo que se ha introducido no es un número.

```

/* Procedimientos de Usuario */
int main(int argc, char *argv[]){
    if(argc==2){
        yyin=fopen(argv[1], "r");
        if(yyin == NULL){
            printf("El fichero %s no se puede abrir\n", argv[1]);
            exit(-1);
        }
    }

    else
        yyin=stdin;

    correcto=0;
    incorrecto=0;

    yylex();
    resultados(correcto, incorrecto);

    fclose(yyin);

    return 0;
}

void resultados(int a, int b){
    printf("\nNumeros correctos: %d \nNumeros incorrectos: %d \n", a, b);
}

```

En esta última sección tenemos los procedimientos, el programa principal que se encargará de leer los números de teléfono que introduzcamos por consola de comandos o por un fichero de texto pasado como parámetro al programa principal.

Finalmente vemos la función resultados comentada en la sección de declaraciones.

Para la compilación del código hemos usado un makefile que tiene las reglas necesarias para generar el ejecutable (make), una regla para ejecutar el código con el fichero con números de ejemplo (make exec_fichero), otra regla que ejecuta el programa para introducir números por consola (make exec) y por último una regla que elimina el ejecutable y el código generado por lex.

3. Capturas de pantalla

```
migue@migue:~/Documentos/MC/Práctica 2$ make
lex plantilla.l
gcc lex.yy.c -o ejecutable -ll
migue@migue:~/Documentos/MC/Práctica 2$ make exec_fichero
./ejecutable ejemplo.txt
627754123 es correcto

45 es incorrecto

694123693 es correcto

hola no es un numero

012345678 es incorrecto

Numeros correctos: 2
Numeros incorrectos: 3
migue@migue:~/Documentos/MC/Práctica 2$ make exec
./ejecutable
prueba
prueba no es un numero

789456123
789456123 es incorrecto

66a987456
66a987456 no es un numero

654789
654789 es incorrecto

619478596
619478596 es correcto

^CMakefile:14: recipe for target 'exec' failed
make: *** [exec] Interrupción

migue@migue:~/Documentos/MC/Práctica 2$ make clean
rm -f *.c ejecutable
migue@migue:~/Documentos/MC/Práctica 2$
```