



UNIVERSIDAD
DE SANTIAGO
DE CHILE

Laboratorio II

Fundamentos de la Ciencia de la Computación I

Profesor:

Rodrigo Abarzúa O.

Integrantes:

Benjamín Riveros L.

Vicente Ahumada V.



1 Introducción

En este laboratorio se solicita desarrollar un programa en Lex (Flex software libre) tal que:

1. Que analice un texto de entrada y sustituya dos o más blancos seguidos por un único blanco y dos o más tabuladores por un único tabulador.
2. Que reciba un fichero de texto y una palabra y cuente el número veces que aparece dicha palabra en el fichero.
3. Que borre los comentarios que aparezcan en un fichero de texto (se suponen comentarios de una sola línea que empiezan por el símbolo #).
4. Que imprima un texto tal y como está en el archivo de entrada, pero que cada vez que detecte un ";", imprima el texto que va a continuación en otra línea nueva, además cada vez que en el texto aparezca ñ remplazar por n.
5. Que indique cuántas veces ha detectado un número entero en un fichero de texto.
6. Además que imprima el texto de salida con los cambios de los puntos anteriores, cifrar el texto utilizando la siguiente tabla de substitución.

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r
C	I	S	Q	V	N	P	O	W	A	X	M	I	G	U	P	B	K

s	t	u	v	w	x	y	z	0	1	2	3	4	5	6	7	8	9
L	R	E	Y	D	Z	J	T	2	7	2	0	4	1	2	6	3	1

Por ejemplo, si el texto es: *Hola mundo, hoy es 24/3/2020*, entonces la salida es: *OUMC TRGQU, OUZ VK 79/2/7575*.

2 Detalles de compilación y ejecución del algoritmo

Para el laboratorio se crearon dos programas en Lex (Flex) en los cuales:

1. **lab2_1.lex** contiene las definiciones, reglas y acciones para los primeros 5 puntos de la introducción, este tiene una salida donde imprime la cantidad de veces que ha detectado un número entero y otra salida que imprime el número de apariciones de la palabra que se desea contar.
2. **lab2_2.lex** realiza el último punto, es decir el cifrado del texto mediante la tabla de substitución e imprime este mismo.

Primero hay que compilar en flex y luego en C uno por uno, es decir

flex lab2_1.lex

gcc -o lab2_1.exe lex.yy.c -lfl

flex lab2_2.lex

gcc -o lab2_2.exe lex.yy.c -lfl

Finalmente para ejecutar ambos programas:

./lab2_1.exe "printf" < entrada.txt | tee salida.txt && ./lab2_2.exe < salida.txt

Esto da salida en formato: **Numeros:** [cantidad de números enteros en el fichero "entrada.txt"]

La palabra 'printf' aparece [n] vez/veces en el fichero

Y el texto antes y después de ser codificado

Nota: Se le puede dar como entrada incluso el mismo archivo .tex, .lex o lex.yy.c



3 Algoritmos

3.1 lab2_1.lex

```
%{
#include<stdio.h>
#include<string.h>
int n_cont = 0;
int w_cont = 0;
char word[] = "integral";
%}

word [a-zA-Z]+

%%

[ ]{2,}    { printf(" "); }

[\t]{2,}    { printf("\t"); }

{word} { if(strcmp(yytext, word)==0){w_cont++;} printf("%s", yytext); }

\#[^\n]*    ;

ñ    { printf("n"); }
Ñ    { printf("N"); }
;    { printf("%s\n", yytext); }

[0-9]+    { n_cont++; printf("%s", yytext); }

.|\n    { printf("%s", yytext); }

%%

int main(int argc, char *argv[]) {
if (argc > 1) {
strcpy(word, argv[1]); // Copia el argumento a 'word'
}
yylex();
fflush(stdout);
fprintf(stderr, "Números: %d\n", n_cont);
fprintf(stderr, "La palabra '%s' aparece %d vez/veces en el fichero\n", word, w_cont);
printf("\n");
return 0;
}
```

3.2 lab2_2.lex

```
%%
[aA]    { printf("C"); }
[bB]    { printf("I"); }
[cC]    { printf("S"); }
[dD]    { printf("Q"); }
[eE]    { printf("V"); }
[fF]    { printf("N"); }
[gG]    { printf("F"); }
[hH]    { printf("O"); }
[iI]    { printf("W"); }
[jJ]    { printf("A"); }
[kK]    { printf("X"); }
[lL]    { printf("M"); }
[mM]    { printf("T"); }
[nN]    { printf("G"); }
[oO]    { printf("U"); }
[pP]    { printf("H"); }
[qQ]    { printf("P"); }
[rR]    { printf("B"); }
[sS]    { printf("K"); }
[tT]    { printf("L"); }
[uU]    { printf("R"); }
[vV]    { printf("E"); }
[wW]    { printf("Y"); }
[xX]    { printf("D"); }
[yY]    { printf("Z"); }
[zZ]    { printf("J"); }
[1]     { printf("4"); }
[2]     { printf("7"); }
[3]     { printf("2"); }
[4]     { printf("9"); }
[5]     { printf("1"); }
[6]     { printf("3"); }
[7]     { printf("0"); }
[8]     { printf("6"); }
[9]     { printf("8"); }
[0]     { printf("5"); }

.|\n    { printf(yytext); }
%%

int main(){
yylex();
printf("\n");
return 0;
}
```

4 Conclusion

Se implementaron dos programas Lex para realizar una serie de cambios en un fichero para luego codificar los caracteres y dar como salida la cantidad de números enteros que hay en el texto, la cantidad de apariciones de una palabra a elección y la codificación del fichero. Se programó en Linux usando Lex en lenguaje C y se usó como texto de prueba el mismo archivo .tex para crear este pdf