

Flex

Un generador de Scanners libre

Jose Pablo Vargas Campos

2013116365

Instituto Tecnológico de Costa Rica

Compiladores e Intérpretes

Semestre 2017

Table of contents

1. Introducción
2. Analisis Léxico

Introducción

Introducción

Flex es una herramienta de análisis léxico desarrollada para la generación de Scanners de lenguajes. Su nombre significa "fast lexical analyzer generator". Es la alternativa gratis y open-source a la herramienta "lex".

Scanning

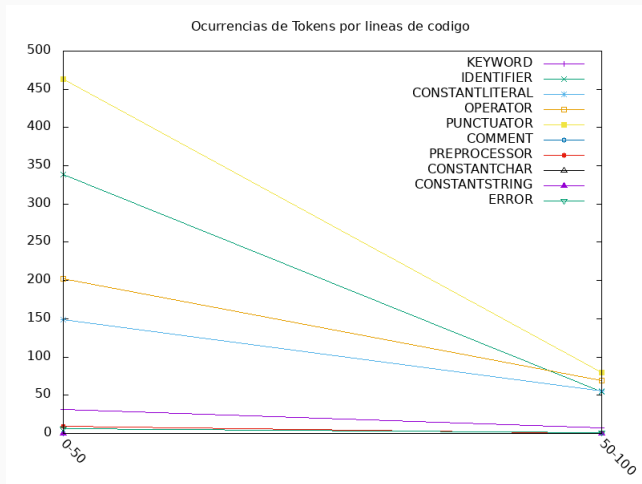
El proceso de Scanning es el proceso por el cual se identifican los diferentes lexemas de un lenguaje. El proceso es tan simple como la ejecución de un Automata Deterministico Finito. Para la generación del Scanner con Flex se utilizan las expresiones regulares, conocidas como 'RegEx', para indicarle a Flex que construya apartir de las expresiones regulares un DFA en C, el cual luego se usa para adquirir los diferentes lexemas del lenguaje que se planea 'Scanear'.

Analisis Léxico

Histograma

A continuación se presenta un histograma el cual indica cuantas veces cada token fue encontrado cada 50 lineas, en el *axis y* se puede ver la cantidad de ocurrencias mientras en el *axis x* se muestra en cual rango de lineas de codigo sucedieron.

Histograma



Codigo fuente

A continuación se presenta el codigo fuente con colores demostrando la división de Tokens.

Syntax Highlighting i

```
#define So long
#define R rand()
#include <math.h>
#include <X11/Xlib.h>
#define T(i,F) ((So long)(i)<<F)
#define O(c,L,i) c*sin(i)+L*cos(i)
#define y(n,L) for(n=0; n<L 3; n++)
#define P(v,L) d=0; y(1,)d+=T(L*1[v],1*20);
#define X(q) q>>10&63|q>>24&4032|q>>38&258048
char J[1<<18]; int G[W*p],_,k,I=W/4+1,w=p/4+1; float C,B,e;

unsigned So long A,n,d, t,h,x,
f,o,r,a,l,L,F,i,s,H=1<<18,b=250,D[1<<14],z[W*p],q
=0x820008202625a0;main(){Display *j=XOpenDisplay(0);Window
```

Syntax Highlighting ii

```
u=XCreateSimpleWindow
(j,RootWindow(j,0),0,0,W,p,1,0,0);XImage
*Y=XCreateImage(j,DefaultVisual(j,0),24
,2,0,(char*)G,W,p,32,0); XEvent M; for(XMapWindow(j,u);
XSelectInput( j,u,1)&&a-
65307; ){ if(!H){ if(XCheckWindowEvent(j,u,1,&M)){
a=XLookupKeysym(&M.xkey,0);*(
a&1?&C:&B)-=(.05
-a/2% 2* .1)*!(a-
1& 4092^ 3920);a+
2&0xfe0^ 0xfc0||(
s=a+2&31 ); }else
{ y(k,p+ ){ F=k%w* 4|k/w; float a[6],S=(F-p /2.) /p;
y(_,W+){ i=_%I*4|_/I; if( F<p&i<W){ o=1; L= i+F*W;if
(1=i&3); else{ l=F&3; o=W ; } h=z[L-o*1]; f =z[L+(4-
```

Syntax Highlighting iii

```
l)*o]; t =F-p/2||i-W/2; r =h^f; if(!l||!t|(int)r|(!
(h- 3&3) &&258063&r>>38)) { float V=(i-W/2. )/p,U=0(
S,1,B),m =32768,Q=m; a[4] =0(-1,S,B); a[3]= 0(U,V,C)
; a[5]=0 (-V,U,C); P((a+3 ),s*42); t||(A=d) ;f=0;y(n
,){float N=a[n+3], E=1024 /fabs(N); b= N<0; float K=
(((q>>20 *n)^^-b)+!b&1023 )/1024.; y(d,)a[d ]=a[d+3]
*E; a[n] =round(a[n]); P( a,K); i=q+d; P(a, 1); e=E*
K; for(; e<m; i+=d){ l=X(i); t=r=l^(l^l-( 1<<6*n))
&63<<6*n ; if(b){ r=l; l=t; } if(J[r])l=r ; if(t=J
[l]){x=( n-1)?(i|i>>40)&1023|i>>8&4190208 |4194304
:i&1023|i>>28&4190208|(b^l==r)<<23; if(h =D[(x>>6
&0xf|x>> 14&0x3f0)+t*768]){ o=h; f=n|l*4| x<<32; m
=e; } if (t==8&e<Q)Q=e; } e+=E; } }b=(255 -((f&3)+
2)%3*51) *(1-m/32768); o=o*b>>8; G[L]=o>> 32<<8|o&
16711935 ; z[L]=3*(Q<=m)|f|b<<56; } else{ d=1*(f<<
```

Syntax Highlighting iv

```
8>>40)+( 4-1)*(h<<8>>40)>>2&16774143; o=D [(d>> 6&
15|d>>14 &1008)+J[(int)h/4]*768]*(b=h>>56 )>>8; G[
L]=o>>32 <<8 | o& 16711935 ; z[L]=(
int) h|d <<32|b<< 56; } } } } q +=A;
XPutImage (j, u+0, DefaultGC (j,J[X(q
)]]=0),Y, 0,0,0,0, W,p); }} else{ L=
--H/768; J[H] =R% 16*(R%4< abs((H>>
6&63)-32 )+abs((H
>>12&63) -32)-3);
i=H &15; F=H %768
>>4; if( L<16){if
(L-1|!(R %3))b=R%
96^255; l=i*i*3+i*81/4&3;
a=L>3?L-8?L-5?9858122:12365733-488848*((i+F/4*4)%8&&F%
4):R%2*5298487:3352537*L*L-14202379*L+19205553;
```

```
if(L==4)if(F<1+18)a=6990400;else
if(1>F-19)b*=0.7; if(L==3){
if((i-1&15)<14&(F-1&15)<14&!(F&16)){ a=12359778; _=7
-i; k=7-F%16; _^=_>>31; k^=k>>31;
b=196-R%32+(k>_?k:_)%3*42;} else{ b*=1+R%2*(.5
-(i&1)); } }
D[H]=(a&16711935|(a&65280)<<24)*(b>>(F>>5))>>8&0xff00ff00ff;
} } }
```

¿Preguntas?