Analizador Léxico

Ariana Bermúdez,Ximena Bolaños, Dylan Rodríguez

Instituto Tecnológico de Costa Rica

May 24, 2017

Análisis Sintáctico

Se hizo un analizador sintáctico con la ayuda de la herramienta de Bison, para el lenguaje C y que corre en C, este analizador trabaja en conjunto con Flex, para tomar los tokens que este le otorga y revisar con las gramáticas que les sean ingresadas.

Bison

jaajaj

Código Preprocesado (Sin Pretty Print)

```
char * lie ;
double time , me = ! 0XFACE ,
not ; int rested , get , out ;
main ( ly , die ) char ly , * * die ; {
signed char lotte ,
dear ; ( char ) lotte -- ;
for ( get = ! me ; ; not ) {
1 - out & out ; lie ; {
char lotte , my = dear ,
* * let = ! ! me * ! not + ++ die ;
( char * ) ( lie =
"The gloves are OFF this time, I detest you, snot\n\0sed GEEK!" ) ;
do { not = * lie ++ & 0xF00L * ! me ;
```

Código Preprocesado (Sin Pretty Print)

```



 (char *) lie -  1 * ! (not = atoi (let
[get - me ?
(char) lotte -
(char) lotte : my - *  (char *) lie -  -
'I' - *  (char *) lie -  - 'U' -
'I' - (long) - 4 - 'U' ] ) - ! !
(time = out = 'a' ) ) ; } while (my - dear
&& 'I' - 1l - get - 'a' ) ; break ; } }
(char) * lie ++ ;
(char) * lie ++ , (char) * lie ++ ; hell : 0 , (char) * lie ;
get * out * (short) ly - 0 - 'R' - get - 'a' ^ rested ;
do
```

Código Preprocesado (Sin Pretty Print)

```
{ auto * eroticism ,
that ; puts ( * ( out
- 'c'
- ( 'P' - 'S' ) + die + - 2 ) ) ; } while ( ! "you're
    at it" ) ;
for ( * ( ( char * ) & lotte ) ^=
( char ) lotte ; ( ( char * ) lie - ly ) [ ( char )
    ++ lotte +
! ! 0xBABE ] ; ) { if ( 'l' - lie [ 2 + ( char ) lotte
    ] ) { 'l' - 1l * * * die ; }
else { if ( 'l' * get * out * ( 'l' - 1l * * die [ 2 ]
    ) ) * ( ( char * ) & lotte ) -=
'4' - ( 'l' - 1l ) ; not ; for ( get = !
get ; ! out ; ( char ) * lie & 0xD0 - ! not ) return !
    !
( char ) lotte ; }
( char ) lotte ;
do { not * putchar ( lie [ out
*
```

Código Preprocesado (Sin Pretty Print)

```
! not * ! ! me + ( char ) lotte ] ) ;
not ; for ( ; ! 'a' ; ) ; } while (
 ( char * ) lie - ( char * ) lie ) ; {
register this ; switch ( ( char ) lie
[ ( char ) lotte ] - 1 * ! out ) {
char * les , get = 0xFF , my ; case ' ' :
* ( ( char * ) & lotte ) += 15 ; ! not + ( char ) * lie
      * 's' ;
this + 1 + not ; default : 0xF + ( char * ) lie ; } } }
get - ! out ;
if ( not -- )
goto hell ;
exit ( ( char ) lotte ) ; }
```