

Analizador Léxico

Proyecto 1

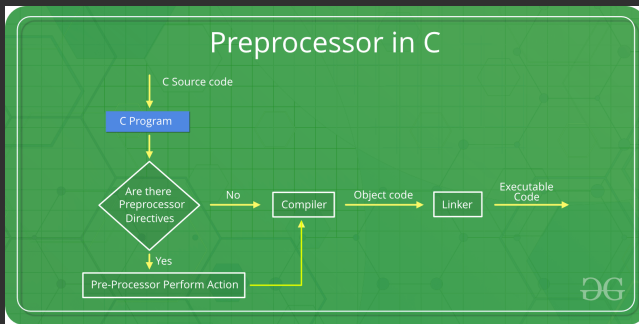
José Pablo Fernández Cubillo
Roberto Vidal Patiño

Tecnológico de Costa Rica
Compiladores e Intérpretes

I Semestre, 2021

Scanning y Herramienta Flex

El proceso de scanning consiste en tomar el código fuente, pasarlo por el preprocesador y luego el archivo generado por el preprocesador es el que se mete en el scanner.



Scanning y Herramienta Flex

El uso de la herramienta flex es relativamente sencillo. Primero se debe definir los diferentes tipos de tokens y se coloca en un archivo header, desde flex se puede importar este archivo. Después, se pueden definir algunas expresiones regulares si se necesitan para identificar algún tipo de token. Luego, se coloca la expresión regular o el propio string y el código del token que debe retornar. Generalmente se usa junto a Bison, el cual sería el parser, pero en este caso como solo es el scanner, se utilizó solo flex.



Programa Fuente

Formato de tokens

- 1 Palabras Reservadas
- 2 Operadores
- 3 String
- 4 Constantes
- 5 Caracteres Especiales
- 6 Identificadores
- 7 Errores Léxicos

Programa Fuente

```
float poly ( float a ( ) , int , float ) ;
```

```
int main ( )
```

```
{
```

```
    float x , a ( 10 ) , y1 ;
```

```
    int deg , i ;
```

```
    printf ( "Enter the degree of polynomial equation: " ) ;
```

```
    scanf ( " % d" , & deg ) ;
```

```
    printf (
```

```
    "Enter the value of x for which the equation is to be evaluated: " ) ;
```

```
    scanf ( " % f" , & x ) ;
```

Programa Fuente

```
for ( i = 0 ; i <= deg ; i ++ )  
{  
    printf ( "Enter the coefficient of x to the power % d: " , i )  
;  
    scanf ( " % f" , & a ( i ) ) ;  
}  
  
y1 = poly ( a , deg , x ) ;  
  
printf ( "  
"The value of polynomial equation for the value of x = % .2f is: % .2f"  
 , x , y1 ) ;
```

Programa Fuente

```
    return 0 ;  
}  
  
float poly ( float a ( ) , int deg , float x )  
{  
    float p ;  
    int i ;  
  
    p = a ( deg ) ;  
  
    for ( i = deg ; i >= 1 ; i -- )
```


Programa Fuente

```
{  
    p = ( a ( i - 1 ) + x * p ) ;  
}  
  
return p ;  
}
```

The End