

Proyecto 1: Analizador Lexico

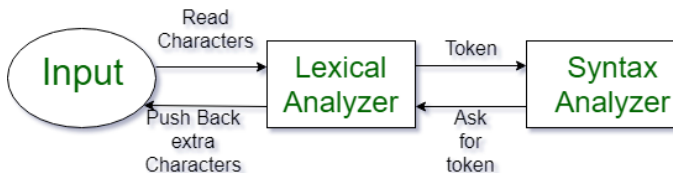
Esteban Durán David Hernández Diego Mendez

Instuto Tecnológico De Costa Rica
Compiladores e Intérpretes

27-04-2022

Scanning and Flex

El proceso de scanning es una fase de la compilación la cual se encarga de tomar el código fuente de entrada (el cual ya pasó por la etapa de preprocesado) y generar tokens que sean representativos del código escrito en este. Cada uno de estos tokens está compuesto por uno o más símbolos que lo representan, este conjunto se le conoce como "Lexema".



Scanning y Flex

Flex genera este scanner de manera sencilla. Este procesa cada token de un archivo de entrada donde uno a uno se extraen mediante una función propia de Flex. Este conjunto de tokens por lo general siguen una string específica o un caracter que los identifique. Si un token lo requiere, se pueden identificar reglas mediante expresiones regulares que ayuden a extraerlo específicamente. Esto también permite crear funciones dentro del mismo scanner que permita manipular esta información según la funcionalidad del scanner.



Programa Fuente I

```
int palindrome ( int num ) {  
    int reversed = 0, remainder, original ;  
    original = num ;  
    while ( num != 0 ) {  
        remainder = num % 10 ;  
        reversed = reversed * 10 + remainder ;  
        num / 10 ;  
    }  
    if ( original == reversed )  
        printf ( "%d is a palindrome." , original ) ;  
    else  
        printf ( "%d is not a palindrome." , original ) ;  
}
```

Programa Fuente II

```
return 0 ;  
}  
  
void mi_prueba ( ) {  
    palindrome ( PI2 ) ;  
    reverseNumber ( PI ) ;  
}  
  
int palindrome ( int num ) {  
    int reversed = 0 , remainder , original ;  
    original = num ;  
    while ( num != 0 ) {  
        remainder = num % 10 ;  
        reversed = reversed * 10 + remainder ;  
    }
```

Programa Fuente III

```
num / 10 ;  
}  
if ( original == reversed )  
printf ( "%d is a palindrome." , original ) ;  
else  
printf ( "%d is not a palindrome." , original ) ;  
return 0 ;  
}  
void mi_prueba ( ) {  
    palindrome ( PI2 ) ;  
    reverseNumber ( PI ) ;  
}
```

Programa Fuente IV

```
int reverseNumber ( int num ) {  
    int reverse = 0 , remainder ;  
    printf ( "Enter an integer: " ) ;  
    scanf ( "%d" , & num ) ;  
    while ( num != 0 ) {  
        remainder = num % 10 ;  
        reverse = reverse * 10 + remainder ;  
        num / 10 ;  
    }  
    printf ( "Reversed number = %d" , reverse ) ;  
    return 0 ;  
}
```

Programa Fuente V

```
int main ( ) {  
    printf ( "Valor de PI es: %i ", PI ) ;  
    mi_prueba ( ) ;  
    return 0 ;  
}
```