

UNIVERSIDAD INTERAMERICANA DE PANAMA

INFORME DE ANALIZADOR LEXICO Y SINTACTICO

LEONARDO ESQUEDA

CLARA BALANTA

LUIS RODRIGUEZ

ALEXANDER MORGAN

22 AGOSTO 2018

INFORME DE COMPILADOR LEXICO Y SINTACTICO

OBJETIVOS

El objetivo del proyecto es el diseño e implementación de un analizador léxico-sintáctico para un lenguaje de programación de alto nivel.

adquirir una visión general de las técnicas de implementación de traductores de lenguaje de programación. En esta asignatura se pone en juego, desde un punto de vista aplicando los conocimientos adquiridos a lo largo de la carrera proporciona un buen ejercicio de síntesis sobre la teoría los métodos y las técnicas aprendidas en programación tales como los compiladores sus fases y cada uno de sus componentes que se proporcionan a lo largo de este proceso.

Conocer el problema que debe resolver un compilador. Su complejidad, así como las facetas que lo conforman. Aprender a implementar programas en código objeto, para después aprender a traducir códigos de un lenguaje de alto nivel a código objeto.

Conocer el funcionamiento del analizador léxico, sintáctico el tipo de gramáticas que debe reconocer y la forma en la que se puede implementar.

Entender las relaciones de seteo con la tabla de símbolo y la gestión de errores

LENGUAJE

Este programa lo realizamos en c++

ARQUITECTURA

La arquitectura del programa se basa en el programa principal el cual invoca el analizador lexico mediante la función `isalpha` comprueba si el carácter es alfabético; en la localización estándar "C", es equivalente a (`isupper` `islower`). En algunas localizaciones, pueden existir caracteres adicionales para los que `isalpha` sea verdadero

`isdigit` comprueba si el carácter es un dígito (de 0 a 9).

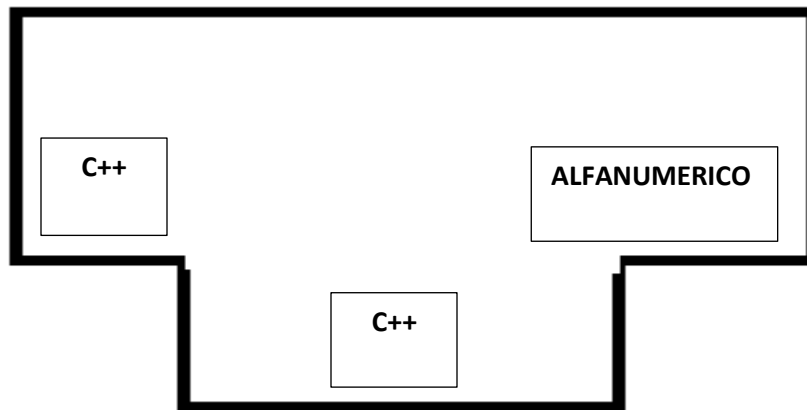
`ispunct` comprueba si el carácter es imprimible sin ser un espacio ni un carácter alfanumérico; o sea, si es un signo de puntuación.

`isspace` comprueba si el carácter es uno de los de espacios en blanco. En las localizaciones "C" y "POSIX" éstos son: espacio, salto de página (`\f`), salto de línea (`\n`), retorno de carro (`\r`), tabulador horizontal (`\t`), y tabulador vertical (`\v`). y le pasa el programa a analizar directamente al analizador léxico, esto pues el analizador sintáctico pide el siguiente token al lexer mediante la función `yylex()`. Si se da algún error en el reconocimiento de tokens, el analizador léxico se encargara de informar dicho error. El caso del que error sea de índole sintáctico será el parser el cual lo informe. Y finalmente si el programa es correcto será el programa principal el que despliegue el éxito de la revisión.

DIAGRAMA DE FLUJO



DIAGRAMA T



DESCRIPCION DE LA UTILIDAD

Este programa analiza una serie de caracteres alfanuméricos por medio de un analizador léxico y sintáctico, los organiza en una tabla de símbolo, suma los números introducidos

DESCRIPCION DE LA TABLA DE SIMBOLO

en la tabla de símbolo se utilizaron caracteres alfanuméricos para la expresión de los analizadores

[!,",#,\$,%,&,/,(),=,?,|,;,*,Ñ,[,:;,_,-,.,~,^,+]* /

MANEJO DE ERRORES

Análisis Léxico Toma el código fuente modificado preprocesadores de idioma que se escriben en la forma de las sentencias. El analizador léxico se rompe estas sintaxis en una serie de fichas, quitando cualquier espacio en blanco o los comentarios en el código fuente.

Si el analizador léxico encuentra un token no válido, se genera un error. El analizador léxico trabaja en estrecha colaboración con la sintaxis analizador. Lee caracteres corrientes desde el código fuente, comprueba de tokens, y pasa los datos a la sintaxis analizador cuando que exige.