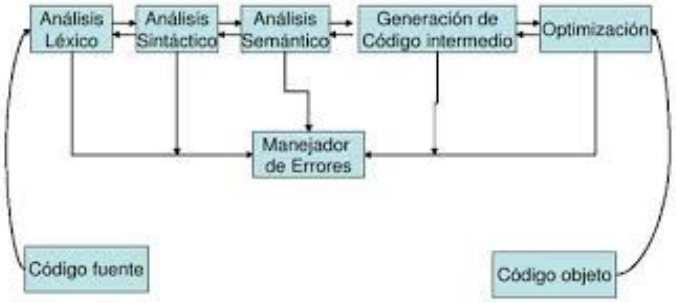
Miércoles 30 julio 2014
Semana 2
Compiladores e interpretes

Diana Arce González. Carne 201216132

Etapas del proceso de compilación

Esta compuesto por 4 etapas,cada una de ellas recibe una entrada y produce una salida.



Análisis léxico

Entrada:programa con código fuente.

Proceso : se agrupan en componentes léxicos (tokens), que son secuencias de caracteres que tienen un significado.

Espacios,lineas en blanco y comentarios se eliminan del programa fuente. Se verifica que los símbolos del lenguaje (palabras reservadas,operadores) se han escrito bien.

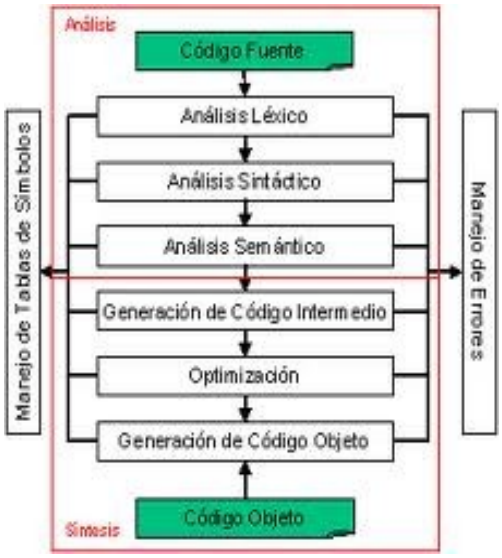
Salida: lista de tokens,

Análisis sintáctico

Entrada: lista de tokens,

Se agrupan jerárquica mente en frases gramaticales que el compilador utiliza para sintetizar la salida. Se comprueba si lo obtenido de la fase anterior es sin tácticamente correcto (obedece a la gramática del lenguaje).

Salida: árbol de análisis sintáctico.



Análisis semántico

Entrada: árbol de análisis sintáctico.

Revisa el programa fuente para tratar de encontrar errores semánticos y reúne la información sobre los tipos para la fase posterior de generación de código. Identificar los operadores y operandos de expresiones y proposiciones.

Aquí, el compilador verifica si cada operador tiene operandos permitidos por la especificación del lenguaje fuente.

Salida: árbol análisis semántico

Fase de síntesis

Consiste en generar el código objeto equivalente al programa fuente. Sólo se genera código objeto cuando el programa fuente está libre de errores de análisis, lo cual no quiere decir que el programa se ejecute correctamente, ya que un programa puede tener errores de concepto o expresiones mal calculadas.

Generación de código intermedio

Después de los análisis sintáctico y semántico, algunos compiladores generan una representación intermedia explícita del programa fuente. Se puede considerar esta representación intermedia como un programa para una máquina abstracta. Esta representación intermedia debe tener dos propiedades importantes; debe ser fácil de producir y fácil de traducir al programa objeto.

Optimización de código

La fase de optimización de código consiste en mejorar el código intermedio, de modo que resulte un código máquina más rápido de ejecutar.

Componentes léxicos o tokens

Cuando un analizador léxico reúne los caracteres en un token, generalmente representa el token de manera simbólica, es decir, como un valor de un tipo de datos enumerado que representa el conjunto de tokens del lenguaje fuente. En la mayoría de los lenguajes el analizador léxico sólo necesita generar un token a la vez. En este caso se puede utilizar una variable global simple para mantener la información del token.

Árbol sintáctico

Si el analizador sintáctico genera un árbol sintáctico, por lo regular se construye como una estructura estándar basada en un puntero que se asigna de manera dinámica a medida que se efectúa el análisis sintáctico. El árbol entero puede entonces conservarse como una variable simple que apunta al nodo raíz. Cada nodo en la estructura es un registro cuyos campos representan la información recolectada tanto por el analizador sintáctico como, posteriormente, por el analizador semántico. Por ejemplo, el tipo de datos de una expresión puede conservarse como un campo en el nodo del árbol sintáctico para la expresión.

En ocasiones, para ahorrar espacio, estos campos se asignan de manera dinámica, o se almacenan en otras estructuras de datos, tales como la tabla de símbolos, que permiten una

asignación y desasignación selectivas. En realidad, cada nodo del árbol sintáctico por sí mismo puede requerir de atributos diferentes para ser almacenado, de acuerdo con la clase de estructura del lenguaje que represente. En este caso, cada nodo en el árbol sintáctico puede estar representado por un registro variable, con cada clase de nodo conteniendo solamente la información necesaria para ese caso.

Tabla de símbolos

Esta estructura de datos mantiene la información asociada con los identificadores: funciones, variables, constantes y tipos de datos. La tabla de símbolos interactúa con casi todas las fases del compilador: el analizador léxico, el analizador sintáctico o el analizador semántico pueden introducir identificadores dentro de la tabla; el analizador semántico agregará tipos de datos y otra información; y las fases de optimización y generación de código utilizarán la información proporcionada por la tabla de símbolos para efectuar selecciones apropiadas de código objeto.

El proceso de ejecución de un programa escrito en un lenguaje de programación y mediante un compilador tiene los siguientes pasos:

1. Escritura del programa fuente con un editor (programa que permite a una computadora actuar de modo similar a una máquina de escribir electrónica) y guardarlo en un dispositivo de almacenamiento (por ejemplo, un disco).
2. Introducir el programa fuente en memoria.
3. Compilar el programa con el compilador.
4. Verificar y corregir errores de compilación (listado de errores).
5. Obtención del programa objeto.
6. El enlazador (linker) obtiene el programa ejecutable.
7. Se ejecuta el programa y, si no existen errores, se tendrá la salida del programa.