



INGENIERÍA EN SISTEMAS COMPUTACIONALES

Materia:

Lenguaje y automatas

Semestre - Sistema:

5 semestre– Escolarizado.

Producto Académico:

1.4 Estructura de un traductor 1.5 Fases de un compilador

Presenta(n):

Itzel diego hernandez

Nombre del maestro:

felix efrain corro isla

Introducción

Un traductor o compilador es un software encargado de convertir un programa escrito en un lenguaje de programación de alto nivel a un lenguaje más bajo, que pueda ser entendido y ejecutado por la máquina. Este proceso se realiza en diferentes fases o etapas, cada una con su función específica.

Objetivo

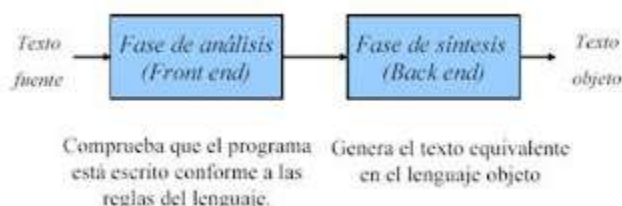
El objetivo de un compilador es transformar el código fuente en un lenguaje de programación (como C++, Java, Python, entre otros) en un lenguaje objeto, que pueda ser ejecutado directamente por la máquina. Para lograr esto, el compilador debe realizar una serie de pasos y análisis del código fuente, asegurándose de que sea válido y generando un código objeto eficiente.

Estructura de un traductor

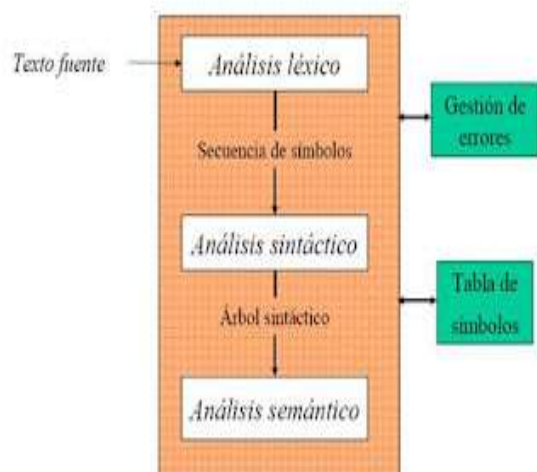
Un traductor es un programa que tiene como entrada un texto escrito en un lenguaje (lenguaje fuente) y como salida produce un texto escrito en un lenguaje (lenguaje objeto) que preserva el significado de origen. Ejemplos de traductores son los ensambladores y los compiladores.



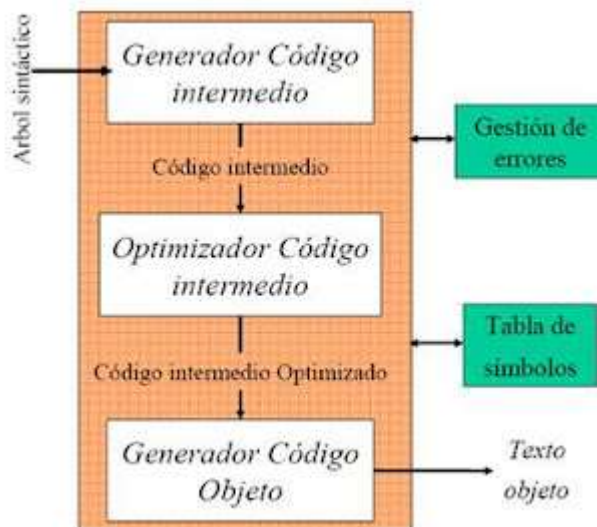
En el proceso de traducción se identifican dos fases principales:



Fase de análisis



Fase de Síntesis



1.4.1 ensambladores.

El programa ensamblador es el programa que realiza la traducción de un programa escrito en ensamblador a lenguaje máquina. Esta traducción es directa e inmediata, ya que las instrucciones en ensamblador no son más que nemotécnicos de las instrucciones máquina que ejecuta directamente la CPU.

Tipos de ensambladores

Podemos distinguir entre tres tipos de ensambladores:

Ensambladores básicos. Son de muy bajo nivel, y su tarea consiste básicamente en ofrecer nombres simbólicos a las distintas instrucciones.

Ensambladores modulares, o macro ensambladores. Descendientes de los ensambladores básicos. Hacen todo lo que puede hacer un ensamblador, y además proporcionan una serie de directivas para definir e invocar macroinstrucciones.

Ensambladores modulares 32-bits o de alto nivel. Son ensambladores que aparecieron como respuesta a una nueva arquitectura de procesadores de 32 bits, realizan la misma tarea que los anteriores, permitiendo también el uso de macros, permiten utilizar estructuras de programación más complejas propias de los lenguajes de alto nivel.

1.4.2 compiladores.

Un compilador es un programa informático que traduce un programa escrito en un lenguaje de programación a otro lenguaje de programación, es decir programa que permite traducir el código fuente de un programa en lenguaje de alto nivel, a otro lenguaje de nivel inferior (lenguaje máquina). Generando un programa equivalente a capaz de interpretar.

Estructura de un Compilador.



Cualquier compilador debe realizar dos tareas principales: análisis del programa a compilar y síntesis de un programa en lenguaje máquina. Para el estudio de un compilador, es necesario dividir su trabajo en fases. Cada fase representa una transformación al código fuente para obtener el código objeto. En cada una de las fases se utiliza un administrador de la tabla de símbolos y un manejador de errores.

Componentes en que se divide un compilador:

Análisis Léxico. En esta fase se lee los caracteres del programa fuente y se agrupan en cadenas que representan los componentes léxicos. A la secuencia de caracteres que representa un componente léxico se le llama lexema (o con su nombre en inglés token).

Análisis Sintáctico. Los componentes léxicos se agrupan en frases gramaticales que el compilador utiliza para sintetizar la salida.

Análisis Semántico. Intenta detectar instrucciones que tengan la estructura sintáctica correcta, pero que no tengan significado para la operación implicada.

Generación de código Intermedio. Se puede considerar esta operación intermedia como un subprograma para una máquina abstracta, a esta representación debe tener dos propiedades importantes: debe ser fácil de producir y fácil de traducir al programa objeto.

Optimización de Código. Se trata de mejorar el código intermedio, de modo que resulte un código de máquina más rápido de ejecutar.

Generación de Código. Esta constituye la fase final de un compilador. Administrador de la tabla de símbolos. Se encarga de manejar los accesos a la tabla de símbolos, en cada una de las etapas de compilación de un programa.

Manejador de errores. Es posible encontrar errores. De esta forma podrán controlarse más eficientemente los errores encontrados en cada una de las fases de la compilación de un programa.

1.4.3 interpretes.

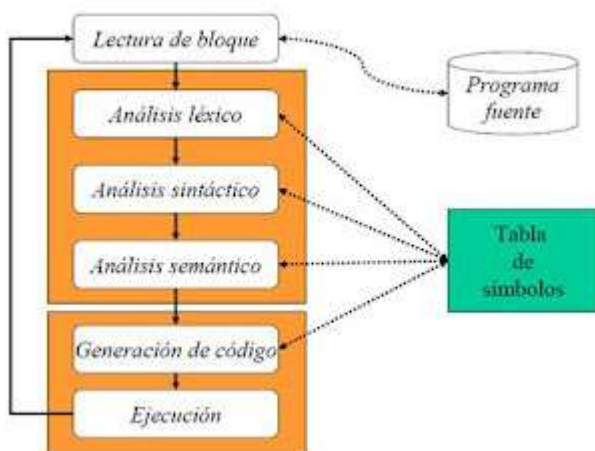
Los intérpretes realizan normalmente dos operaciones:

Traducen el código fuente a un formato interno.

Ejecutan o interpretan el programa traducido al formato interno.

La primera parte del intérprete se llama a veces "el compilador", aunque el código interno que genera no es el lenguaje de la máquina, ni siquiera lenguaje simbólico, ni tampoco un lenguaje de alto nivel.

Estructura



Fases de un compilador

Un compilador típicamente opera en fases , cada una lleva a cabo una tarea sobre el programa fuente. Las primeras tres fases suelen agruparse en una sola fase llamada fase de análisis y las últimas tres en una llamada fase de síntesis. La fase de análisis y el modulo de manejo de errores se describen posteriormente en este mismo capítulo. La fase de síntesis no es relevante en el contexto de un lenguaje multibase de datos, ya que este sigue un enfoque diferente que el de los lenguajes tradicionales, por esta razón solo se menciona. Muchas herramientas de software que manipulan programas fuente realizan primero algún tipo de análisis, entre estas se encuentran los editores de estructuras, impresoras estéticas, verificadores estáticos y los interpretes¹. Estructura de un Compilador. Cualquier compilador debe realizar dos tareas principales: análisis del programa a compilar y síntesis de un programa en lenguaje maquina que, cuando se ejecute, realizara correctamente las actividades descritas en el programa fuente. Para el estudio de un compilador, es necesario dividir su trabajo en fases. Cada fase representa una transformación al código fuente para obtener el código objeto. La siguiente figura representa los componentes en que se divide un compilador. Las tres primeras fases realizan la tarea de análisis, y las demás la síntesis. En cada una de las fases se utiliza un administrador de la tabla de símbolos y un manejador de errores.

Análisis Léxico.

En la fase de análisis léxico se leen los caracteres del programa fuente y se agrupan en cadenas que representan los componentes léxicos. Cada componente léxico es una secuencia lógicamente coherente de caracteres relativa a un identificador, una palabra reservada, un operador o un carácter de puntuación. A la secuencia de caracteres que representa un componente léxico se le llama lexema (o con su nombre en inglés token). En el caso de los identificadores creados por el programador no solo se genera un componente léxico, sino que se genera otro lexema en la tabla de símbolos.

Análisis Sintáctico.

En esta fase, los componentes léxicos se agrupan en frases gramaticales que el compilador utiliza para sintetizar la salida. Análisis Semántico.

La fase de análisis semántico se intenta detectar instrucciones que tengan la estructura sintáctica correcta, pero que no tengan significado para la operación implicada.

Análisis Semántico: Este análisis es más difícil de formalizar, determina el tipo de los resultados intermedios, comprobar que los argumentos que tienen un operador pertenecen al conjunto de operadores posible, y si son compatibles entre sí.

Generación de código Intermedio.

Algunos compiladores generan una representación intermedia explícita del programa fuente, una vez que se han realizado las fases de análisis. Se puede considerar esta operación intermedia como un subprograma 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.
En esta fase se trata de mejorar el código intermedio, de modo que resulte un código de máquina más rápido de ejecutar.
Generación de Código.
Esta constituye la fase final de un compilador. En ella se genera el código objeto que por lo general consiste en código en lenguaje máquina (código relocizable) o código en lenguaje ensamblador.

“Administrador de la tabla de símbolos.” Una tabla de símbolos es una estructura de datos que contiene un registro por cada identificador. El registro incluye los campos para los atributos del identificador. El administrador de la tabla de símbolos se encarga de manejar los accesos a la tabla de símbolos, en cada una de las etapas de compilación de un programa.

Conclusión

un traductor es una herramienta que permite convertir un código fuente de un lenguaje de programación a otro, mientras que un compilador es un tipo de traductor que realiza esta conversión de manera completa, pasando por diversas fases.

La estructura de un traductor consta de un analizador léxico, un analizador sintáctico, un analizador semántico y un generador de código intermedio. El analizador léxico se encarga de identificar las unidades léxicas, como palabras clave y símbolos, el analizador sintáctico verifica la estructura gramatical del código, el analizador semántico realiza comprobaciones de coherencia y el generador de código intermedio traduce el código a un lenguaje intermedio. Además, se puede incluir una fase de optimización para mejorar el rendimiento del programa.

Por otro lado, las fases de un compilador son la fase de análisis léxico, la cual se encarga de identificar las unidades léxicas, la fase de análisis sintáctico que verifica la estructura gramatical del código, la fase de análisis semántico que realiza comprobaciones de coherencia, la fase de generación de código intermedio que traduce el código a un lenguaje intermedio y finalmente, la fase de generación de código que convierte el código intermedio en código ejecutable.

En resumen, tanto un traductor como un compilador son herramientas esenciales en la programación, permitiendo la conversión de código entre distintos lenguajes y asegurando que el código sea válido y ejecutable.

Bibliografías

<https://ing-radichv-pmz-estl.blogspot.com/2009/09/14-traductor-y-su-estructura.html>

<https://www.buenastareas.com/ensayos/Fases-De-Un-Compilador/45439519.html>