

Unidad I. Conceptos Introdutorios. Compiladores.

Que es un Compilador?

Un compilador acepta programas escritos en un lenguaje de alto nivel y los traduce a otro lenguaje, generando un programa equivalente independiente, que puede ejecutarse tantas veces como se quiera. Este proceso de traducción se conoce como compilación.

Los programas interpretados suelen ser más lentos que los compilados, pero los intérpretes son más flexibles como entornos de programación y depuración. Comparando su actuación con la de un ser humano, un compilador equivale a un traductor profesional que, a partir de un texto, prepara otro independiente traducido a otra lengua, mientras que un intérprete corresponde al intérprete humano, que traduce de viva voz las palabras que oye, sin dejar constancia por escrito.

Partes de un compilador.

Normalmente los compiladores están divididos en dos partes:

Front End: es la parte que analiza el código fuente, comprueba su validez, genera el árbol de derivación y rellena los valores de la tabla de símbolos. Esta parte suele ser independiente de la plataforma o sistema para el cual se vaya a compilar.

Back End: es la parte que genera el código máquina, específico de una plataforma, a partir de los resultados de la fase de análisis, realizada por el *Front End*.

Esta división permite que el mismo *Back End* se utilice para generar el código máquina de varios lenguajes de programación distintos y que el mismo *Front End* que sirve para analizar el código fuente de un lenguaje de programación concreto sirva para la generación de código máquina en varias plataformas distintas. El código que genera el *Back End* normalmente no se puede ejecutar directamente, sino que necesita ser enlazado por un programa enlazador (*linker*).

Tipos de compiladores

Esta taxonomía de los tipos de compiladores no es excluyente, por lo que puede haber compiladores que se adscriban a varias categorías:

Compiladores cruzados: generan código para un sistema distinto del que están funcionando.

Compiladores optimizadores: realizan cambios en el código para mejorar su eficiencia, pero manteniendo la funcionalidad del programa original.

Compiladores de una sola pasada: generan el código máquina a partir de una única lectura del código fuente.

Compiladores de varias pasadas: necesitan leer el código fuente varias veces antes de poder producir el código máquina.

Compiladores Incrementales: generan el código de máquina a medida que van compilando el código fuente.

AutoCompiladores: el compilador está escrito en el mismo lenguaje de alto nivel que va a compilar.

MetaCompilador: es un compilador de compiladores y se refiere a un programa que recibe como entrada las especificaciones del lenguaje para el que se desea obtener un compilador y genera como salida el compilador para ese lenguaje.

Proceso de creación de un compilador.

En las primeras épocas de la informática, el software de los compiladores era considerado como uno de los más complejos existentes.

Los primeros compiladores se realizaron programándolos directamente en lenguaje máquina o en ensamblador. Una vez que se dispone de un compilador, se pueden escribir nuevas versiones del compilador (u otros compiladores distintos) en el lenguaje que compila ese compilador.

Actualmente existen herramientas que facilitan la tarea de escribir compiladores ó intérpretes informáticos. Estas herramientas permiten generar el esqueleto del analizador sintáctico a partir de una definición formal del *lenguaje de partida*, especificada normalmente mediante una gramática formal, dejando únicamente al programador del compilador la tarea de programar las acciones semánticas asociadas.