

El Compilador y Sus Fases



Es un programa que puede leer un programa en un lenguaje (el lenguaje fuente) y traducirlo en un programa equivalente en otro lenguaje (el lenguaje destino); para que esto se lleve a cabo se debe pasar por varias fases que se describen a continuación.

////////// Análisis del lenguaje

1. Léxico

Lee el flujo de caracteres que componen el programa y los agrupa en secuencias significativas o lexemas, y para cada lexema produce una salida o token.

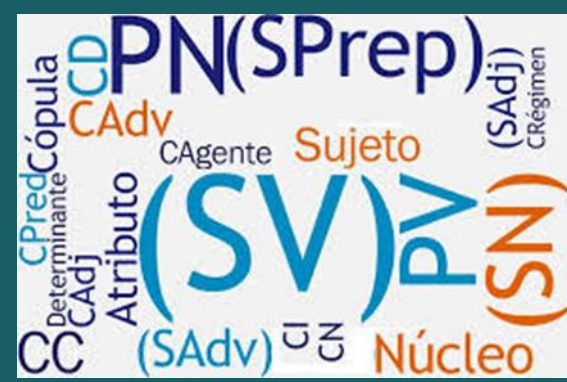


2. sintáctico

Utiliza los primeros componentes de los tokens para crear una representación intermedia en forma de árbol que describa la estructura gramatical del flujo de tokens.

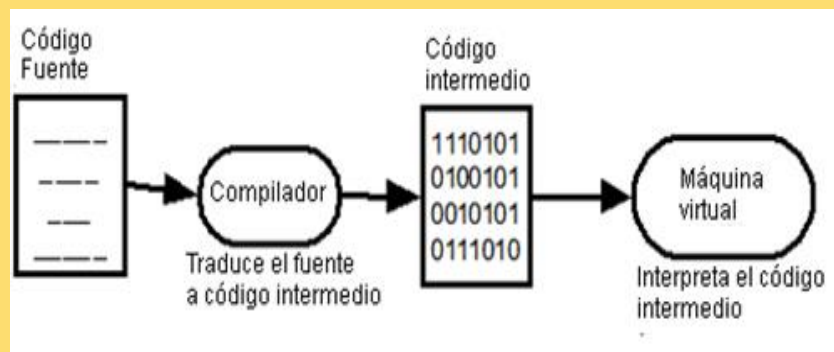
3. Semántico

Utiliza el árbol sin tático y la información en la tabla de símbolos para comprobar la consistencia semántica del programa fuente con la definición del lenguaje.



////////// Generación de Código intermedio

En el proceso de traducir un programa fuente a código destino, un compilador puede construir una o más representaciones intermedias, las cuales pueden tener una variedad de formas. Los árboles sin táticos son una forma de representación intermedia; por lo general, se utilizan durante el análisis sin tático y semántico. Muchos compiladores generan un nivel bajo explícito, o una representación intermedia similar al código máquina, que podemos considerar como un programa para una máquina abstracta. Esta representación intermedia debe tener dos propiedades importantes: debe ser fácil de producir y fácil de traducir en la máquina destino.



Optimización de Código //////////



Trata de mejorar el código intermedio, de manera que se produzca un mejor código destino. pero pueden lograrse otros objetivos, como un código más corto, o un código de destino que consuma menos poder de procesamiento.

////////// Administración de la tabla de símbolos

Es una estructura de datos que contiene un registro para cada nombre de variable, con campos para los atributos del nombre. La estructura de datos debe diseñarse de tal forma que permita al compilador buscar el registro para cada nombre, y almacenar u obtener datos de ese registro con rapidez.

Alfabeto Fonético Internacional - Discretivo				
Serás	ʃ	Ma	mas	mas
Seras	s	Masa	mas	mas
Aspirada	tʃ	Aspirada	tʃ	Aspirada
Suspirada	tʃ	Aspirada	tʃ	Aspirada
Re-	tʃ	Aspirada	tʃ	Aspirada
Ligam-	tʃ	Aspirada	tʃ	Aspirada
Dental	tʃ	Aspirada	tʃ	Aspirada
Apical	tʃ	Aspirada	tʃ	Aspirada
Laminal	tʃ	Aspirada	tʃ	Aspirada

////////// Generación de código

Recibe como entrada una representación intermedia del programa fuente y la asigna al lenguaje destino. Si el lenguaje destino es código máquina, se seleccionan registros o ubicaciones (localidades) de memoria para cada una de las variables que utiliza el programa. Después, las instrucciones intermedias se traducen en secuencias de instrucciones de máquina que realizan la misma tarea.



El agrupamiento de fases en pasadas

la organización lógica de un compilador. En una implementación, las actividades de varias fases pueden agruparse en una pasada, la cual lee un archivo de entrada y escribe en un archivo de salida



Herramientas de construcción de compiladores

Estas herramientas utilizan lenguajes especializados para especificar e implementar componentes específicos, y muchas utilizan algoritmos bastante sofisticados. Las herramientas más exitosas son las que ocultan los detalles del algoritmo de generación y producen componentes que pueden integrarse con facilidad al resto del compilador.

