
ANÁLISIS LÉXICO
Es la primera fase del compilador. Su función es leer caracteres del código fuente y agruparlos en secuencias significativas llamas lexemas, a partir de cada lexema, genera un token en forma nombre-atributo, y luego pasa al análisis sintáctico
ANÁLISIS SINTÁCTICO
Su función es recibir los tokens generados por el análisis léxico y construir una representación intermedia en forma de árbol que refleje la estructura gramatical del programa. Ej: cada nodo interno representa una operación y los hijos representan los operandos
ANÁLISIS SEMÁNTICO
Su objetivo es verificar que el programa fuente sea semánticamente correcto según las reglas del lenguaje de programación. Utiliza el árbol de sintaxis y la tabla de símbolos para comprobar la coherencia del código
GENERACIÓN DE CÓDIGO INTERMEDIO
Durante la traducción del código fuente, un compilador puede generar una o más representaciones intermedias ates de producir el código final.
GENERACIÓN DE CÓDIGO
Es la fase del compilador que convierte la representación intermedia del programa en código en el lenguaje destino. Esta fase puede incluir asignación de memoria.
OPTIMIZACIÓN DE CÓDIGO
Busca mejorar la representación intermedia para generar un código final más eficiente. Generalmente, optimizar significa hacer el código más rápido, pero también puede reducir su tamaño o consumo de energía
Administración de tabla de símbolos: Una función esencial del compilador es registrar los nombres de variables del programa fuente y almacenar información sobre sus atributos como: Espacio de memoria asignado, tipo de dato, alcance.. Toda esta información se guarda en la tabla de símbolos, una estructura de datos optimizada para almacenar y recuperar información de manera rápida durante la compilación