

Resumen del libro “Data Structures = Programs”

Santiago Valencia Díaz – 1004754681

El proceso de compilación en PL/0 involucra varios pasos esenciales para convertir el código fuente en código máquina ejecutable. Primero, el compilador asocia los códigos de operación y sus parámetros con identificadores específicos utilizando una tabla que incluye atributos para cada identificador. Por ejemplo, si el identificador es una constante, su atributo es su valor; si es una variable, su atributo es su dirección; y si es un procedimiento, sus atributos incluyen la dirección de entrada y el nivel del procedimiento.

Las direcciones de las variables se asignan de forma secuencial, simplificando el proceso en PL/0. Cada vez que se declara una variable, se incrementa un índice de asignación de datos (dx), que se inicializa en 3 al inicio de la compilación de un procedimiento, para incluir tres variables internas obligatorias (RA, DL y SL). El procedimiento `enter` se encarga de calcular y registrar los atributos de los identificadores en la tabla.

La generación del código real es directa gracias a la organización en pila de la máquina PL/0, permitiendo una correspondencia casi directa entre operadores y operandos en el lenguaje fuente y las instrucciones en el código objetivo. El compilador reordena las instrucciones en forma postfija, simplificando la evaluación de expresiones al eliminar la necesidad de paréntesis.

La traducción de declaraciones condicionales y de bucles requiere la generación de instrucciones de salto, cuyo destino puede no conocerse de inmediato. El compilador maneja esto mediante dos enfoques: un esquema de dos pasadas para completar las direcciones de salto, o el uso de una matriz que permite actualizaciones directas (fixup) cuando las direcciones se conocen.

El procedimiento auxiliar `gen` facilita la emisión de instrucciones, ensamblándolas según sus parámetros y actualizando automáticamente el índice de código (ex) para la próxima instrucción.

Aunque el compilador de PL/0 es sencillo, extenderlo para manejar lenguajes más complejos o computadoras más convencionales es considerablemente más difícil debido a la complejidad de organizar las tareas de compilación. Sin embargo, el enfoque básico sigue

siendo válido y valioso para tareas más complejas y ha demostrado ser efectivo en la construcción de grandes compiladores.

En cuanto al código, el compilador PL/0 procesa declaraciones de constantes, variables y procedimientos, y funciones para tareas como obtener símbolos, generar código, realizar pruebas y generar código intermedio, para luego ejecutar el código resultante.