



Tecnológico Nacional de México.

Instituto Tecnológico de Nuevo
León.



Ingeniería en Sistemas Computacionales
Lenguajes Autómatas II

“Generación de Código Objeto”

Trabajo presentado por:

García Hernández Julio César.

#15480089

Guadalupe, Nuevo León a 16 de Noviembre del 2018,

Introducción

En este proyecto se abarcarán 4 temas muy importantes, que son los registros, el lenguaje ensamblador, el lenguaje máquina y la administración de memoria. Todo esto para llegar a la fase final de un compilador que es la generación de código objeto, que por lo general consiste en código de máquina relocalizable o código ensamblador.

El generador de código objeto puede considerarse como la penúltima fase de un compilador, la cual se encarga de tomar como entrada el código intermedio generado por el front-end, y producir código objeto de la arquitectura target para luego entrar en la fase de optimización de código.

Toma como entrada de representación intermedia el programa fuente y produce como salida un programa objeto equivalente.

Capítulo 1: Registros

Las instrucciones que implican operandos en registros son generalmente más rápidas que las de los operandos en memoria. Por lo tanto, utilizar eficientemente los registros es fundamental para generar un buen código. El uso de registros se divide en dos subproblemas:

- Durante la asignación de los registros: se selecciona el conjunto de variables que residirá en los registros en un momento del programa.
- Durante la fase posterior de asignación a los registros, se escoge el registro específico en el que residirá una variable.

Existen varios tipos de registros:

- Registros de segmento
- Registros de apuntadores de instrucciones
- Registros apuntadores
- Registros de propósitos generales
- Registro índice
- Registro de bandera

Capítulo 2: Lenguaje ensamblador

El ensamblador es un traductor de un código de bajo nivel a un código, ejecutable directamente por la máquina para la que se ha generado.

Fue la primera abstracción de un lenguaje de programación, posteriormente aparecieron los compiladores.

Ensambladores

- Ensambladores básicos: Son de muy bajo nivel, y su tarea consiste básicamente en ofrecer nombres simbólicos a las distintas instrucciones, parámetros y cosas tales como los modos de direccionamiento.
- Ensambladores modulares, o macro ensambladores: Descendientes de los ensambladores básicos, fueron muy populares en las décadas de los 50 y los 60, antes de la generalización de los lenguajes de alto nivel. Un macroinstrucción es el equivalente a una función en un lenguaje de alto nivel.

Almacenamiento

Una de las principales ventajas del uso del ensamblador es:

- Que se encarga de administrar de manera transparente para el usuario la creación de memoria, las bifurcaciones y el paso de parámetros.
- Además, nos permite acceder directamente a los recursos de la máquina para un mejor desempeño.

Manejo de la Memoria: Direccionamiento (Interno y Externo)

El manejo de la memoria depende de que procesador tenga la máquina, entre los cuales a continuación se mencionan los siguientes:

- Memoria de Programa
- Memoria Externa de Datos
- Memoria Interna de Datos
- Registros de Funciones Especiales
- Memoria de Bit

Capítulo 3: Lenguaje máquina

El lenguaje máquina sólo es entendible por las computadoras. Se basa en una lógica binaria de 0 y 1, generalmente implementada por mecanismos eléctricos.

En general el lenguaje máquina es difícil de entender para los humanos por este motivo hacemos uso de lenguajes más parecidos a los lenguajes naturales.

Direccionamiento: Es la forma en cómo se accede a la memoria. Recordar que un programa no puede ejecutarse sino se encuentra en memoria principal. La forma de acceder a la memoria depende del microprocesador, pero en general existen dos tipos de direccionamiento: directo e indirecto.

- El direccionamiento directo también recibe el nombre de direccionamiento absoluto y el acceso a las direcciones se hace de manera directa.

- El direccionamiento indirecto también recibe el nombre de direccionamiento relativo y se basa a partir de una dirección genérica, generalmente el inicio del programa. Para acceder a una dirección relativa se suma a la dirección base el número de espacios de memorias necesarias.

Capítulo 4: Administración de memoria

Consiste en determinar la posición de memoria en la que los diferentes símbolos del programa almacenan la información. Depende de la estrategia utilizada para la gestión de memoria, el mecanismo puede variar.

En la mayoría de los lenguajes de programación el uso de punteros no estaba vigilado por lo que se tienen muchos problemas con el uso de memoria. Los lenguajes más recientes controlan el uso de punteros y tienen un programa denominado recolector de basura que se encarga de limpiar la memoria no utilizada mejorando el desempeño.

El administrador de memoria tiene como objetivos:

- Ubicar, reemplazar, cargar y descargar procesos en la memoria principal.
- Proteger la memoria de acceso indeseados (accidentales o intencionados).
- Permitir la compartición de zonas de memoria (indispensable para lograr la cooperación de procesos).

En la administración de memoria las técnicas usadas son las siguientes:

- Partición Fija: La memoria principal se divide en un conjunto de particiones de tamaño fijo durante el inicio del sistema. Un proceso se puede cargar completamente en una partición de tamaño menor o igual.
 - Ventajas. Sencilla de implementar. Poca sobrecarga al SO.
 - Desventajas. Fragmentación interna. Número fijo de procesos activos.
- Partición Dinámica: Las particiones se crean dinámicamente por demanda. Son variables en tamaño y número. Cada proceso se carga completamente en una única partición del tamaño del proceso.
 - Ventajas. No existe fragmentación interna.
 - Desventajas. Fragmentación externa. Se debe compactar la memoria, el compactado toma tiempo.
- Segmentación Simple: Cada proceso y sus datos se dividen en segmentos de longitud variable. Un proceso carga sus segmentos en particiones dinámicas no necesariamente contiguas. Todos los segmentos de un proceso se deben de cargar en memoria. Se diferencia de la partición dinámica en que un proceso puede ocupar más de un segmento.
 - Ventajas. No hay fragmentación interna.
 - Desventajas. Fragmentación externa, pero menor (compactación).
- Memoria Virtual Paginada

- Memoria Virtual Segmentada

Conclusiones

Los compiladores son programas traductores, que leen un fichero de texto que contiene el programa fuente y generan un fichero que contiene el código objeto.

El código objeto no suele tener ningún significativo para los seres humanos, además es diferente para cada ordenador y para cada sistema operativo. Por lo tanto, existen diferentes compiladores para diferentes sistemas operativos y para cada ordenador

Se puede concluir en que el código objeto son un conjunto de instrucciones y datos escritos en un lenguaje de programación que entiende el ordenador directamente, escritos en binario o código máquina. Proviene de la traducción de cierto código fuente, es un fragmento del programa final y es específico de la plataforma de ejecución.

Conceptos

Registros

Registros: Un registro es una memoria de alta velocidad y poca capacidad, integrada en el microprocesador, que permite guardar transitoriamente y acceder a valores muy usados, generalmente en operaciones matemáticas.

Variables: es un espacio en memoria reservado para almacenar un valor que corresponda a un tipo de dato soportado por el lenguaje de programación.

Lenguaje ensamblador

Ensamblador: es un traductor de un código de bajo nivel a un código, ejecutable directamente por la máquina para la que se ha generado.

Lenguaje de programación: es un lenguaje formal diseñado para realizar procesos que pueden ser llevados a cabo por máquinas como las computadoras.

Abstracción: consiste en aislar un elemento de su contexto o del resto de los elementos que lo acompañan.

Compilador: es un programa informático que traduce un programa que ha sido escrito en un lenguaje de programación a un lenguaje común.

Parámetros: es una variable utilizada para recibir valores de entrada en una rutina, subrutina o método.

Direccionamiento: son las diferentes maneras de especificar un operando dentro de una instrucción en lenguaje ensamblador.

Almacenamiento: capacidad de guardar datos que tiene un dispositivo electrónico.

Lenguaje máquina

Lenguaje máquina: sólo es entendible por las computadoras. Se basa en una lógica binaria de 0 y 1, generalmente implementada por mecanismos eléctricos.

Administración de memoria

Memoria: En informática, la memoria es el dispositivo que retiene, memoriza o almacena datos informáticos durante algún período de tiempo.

Punteros: es un objeto del lenguaje de programación, cuyo valor se refiere a otro valor almacenado en otra parte de la memoria del ordenador utilizando su dirección.

Recolector de basura: es un mecanismo implícito de gestión de memoria implementado en algunos lenguajes de programación de tipo interpretado o semiinterpretado.

Bibliografía

- Alfred V, Compiladores principios, técnicas y herramientas. Segunda edición, Aho. Pearson Educación, México, 2008.
- Generación de código objeto, descrita en <http://carlosq4lf.blogspot.mx/>
- Generador de Código Objeto, descrita en http://cidecame.uaeh.edu.mx/lcc/mapa/PROYECTO/libro32/autocontenido/autocon/137_generador_de_codigo_objeto.html
- Generación de Código Objeto, descrita en <http://unidad4generacion.blogspot.mx/>

Reporte

Registros

Los registros son la memoria principal de la computadora, algunos de estos son utilizados para funciones en específico, siendo de dos tipos de propósito general y de uso exclusivo. Hay diferentes tipos de registros entre ellos los punteros que mencione en el resumen.

Lenguaje ensamblador

El lenguaje ensamblador tiene la función de leer un archivo en lenguaje ensamblador y sustituye cada uno de los caracteres por su similar en código máquina. Con esto los programas se hacen fácilmente portables y los cálculos se hacen más fáciles.

Los registros del procesador se emplean para controlar instrucciones en ejecución, manejar direccionamiento de memoria y proporcionar capacidad aritmética.

Lenguaje máquina

El lenguaje maquina realiza un conjunto de operaciones mediante microoperaciones las cuales solo pueden realizar operaciones aritméticas, lógicas y de control.

El lenguaje maquina depende de la arquitectura de la computadora, esto hace que los programas máquina para cierta arquitectura no sean compatibles con otro tipo de arquitectura diferente.

Administración de memoria

La administración de memoria es un proceso que hoy en día es muy importante, de tal modo que su buen o mal uso tiene una acción directa sobre el desempeño de la memoria, involucrando costos y sobre todo afectando a la optimización.