



TECNOLÓGICO NACIONAL DE MÉXICO INSTITUTO TECNOLÓGICO DE NUEVO LEÓN

ING. SISTEMAS COMPUTACIONALES

Lenguajes y autómatas II

Proyecto 4

Generación de código objeto

ALUMNO: Salvador Eliud Carranza de la Rosa

NO. CONTROL: 15480101

CATEDRÁTICO: Juan Pablo Rosas Baldazo

Introducción

A continuación se presenta un resumen de la unidad 4 de la materia Lenguajes y Autómatas II, el cual se pidió como proyecto 4, en él se abordan todos los temas de la unidad “Generación de código objeto”, desglosados por capítulos, en el capítulo 1 se habla de los registros nos dice que los registros son la memoria principal de la computadora. Hay diversos registros de propósito general y otros de uso exclusivo. Algunos registros de propósito general son utilizados para cierto tipo de funciones. En el capítulo 2 se habla del lenguaje ensamblador que es un traductor de un código de bajo nivel a un código objeto que es ejecutable directamente por la máquina para la que se ha generado. El capítulo 3 habla sobre el lenguaje máquina que es el que proporciona poca o ninguna abstracción del microprocesador de un ordenador este solo es entendible por las computadoras y es difícil de entender para los humanos se denomina lenguaje máquina a la serie de datos que la parte física de la computadora o hardware, es capaz de interpretar. Y por último el capítulo 5 que trata sobre la administración de memoria que consiste en determinar la posición de memoria en la que los diferentes símbolos del programa almacenan la información y depende de la estrategia utilizada para la gestión de memoria, el mecanismo puede variar, la administración de la memoria es un proceso hoy en día muy importante, de tal modo que su mal o buen uso tiene una acción directa sobre el desempeño de memoria.

Unidad IV Generación de Código Objeto

El generador de código objeto como lo menciona (Urbina, 2011) transforma el código Intermedio optimizado en código objeto de bajo nivel. Toma código intermedio y genera Código objeto para la máquina considerada. Es la parte más próxima a la arquitectura de la Máquina. Habitualmente, se escriben ``a mano'' desarrollo a medida para cada máquina Específica.

Capítulo 1: Registros

Los registros son la memoria principal de la computadora. Existen diversos registros de propósito general y otros de uso exclusivo. Algunos registros de propósito general son utilizados para cierto tipo de funciones. Existen registros acumuladores, puntero de instrucción, de pila, etc.

Distribución

La distribución es el proceso en el que el programa generado puede ejecutarse en otras máquinas.

Con respecto al ensamblador, la mayoría del direccionamiento se hace relativo para que el programa sea relocizable por un programa llamado cargador, en el caso de programas compilados se necesitan de las librerías, si son estáticas se incluyen en el ejecutable por lo que el programa se hace gráfico, si son dinámicas no pero el programa es más pequeño, debido a la complejidad del software actual se necesitan de asistentes para poder instalar y ejecutar un programa.

Los registros se pueden clasificar de la siguiente forma:

Registros de datos:

AX: Registro acumulador. Es el principal empleado en las operaciones aritméticas.

BX: Registro base. Se usa para indicar un desplazamiento.

CX: Registro contador. Se usa como contador en los bucles.

DX: Registro de datos.

Registros de segmentos:

CS: Registro de segmento de código. Contiene la dirección de las instrucciones del programa.

DS: Registro segmento de datos. Contiene la dirección del área de memoria donde se encuentran los datos del programa.

SS: Registro segmento de pila. Contiene la dirección del segmento de pila. La pila es un espacio de memoria temporal que se usa para almacenar valores de 16 bits (palabras).

ES: Registro segmento extra. Contiene la dirección del segmento extra. Se trata de un segmento de datos adicional que se utiliza para superar la limitación de los 64Kb del segmento de datos y para hacer transferencias de datos entre segmentos.

Registros punteros de pila:

SP: Puntero de la pila. Contiene la dirección relativa al segmento de la pila.

BP: Puntero base. Se utiliza para fijar el puntero de pila y así poder acceder a los elementos de la pila.

Registros índices:

SI: Índice fuente.

DI: Índice destino.

El uso de registros puede dividirse en dos subproblemas:

– Durante la reserva de registros (allocation), se seleccionan el conjunto de variables que vivirá en registros en un punto del programa.

– Durante la (posterior) asignación de registros (assignment), se elige el registro específico para cada variable.

Capítulo 2: Lenguaje ensamblador

El ensamblador (del inglés assembler) 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.

Características

- El programa lee un archivo escrito en lenguaje ensamblador y sustituye cada uno de los códigos mnemotécnicos por su equivalente código máquina.
- Los programas se hacen fácilmente portables de máquina a máquina y el cálculo de bifurcaciones se hace de manera fácil.

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.

Operaciones básicas

(Urbina, 2011) Las operaciones básicas en un lenguaje ensamblador son la suma la resta la multiplicación y la división y Necesitara un poco más de información sobre la arquitectura y SO.

Pero la idea básica es:

--definir qué parámetros tendrá la función.

--hacer el programa, propiamente dicho, en assembler.

Siguiendo la convención de pasaje de parámetros, manejará registros y posiciones de memoria, devolviendo los resultados en donde deba (una posición de memoria, el registro eax, etc.)

Capítulo 3: Lenguaje máquina

Es el que proporciona poca o ninguna abstracción del microprocesador de un ordenador. El lenguaje máquina solo es entendible por las computadoras. Se basa en una lógica binaria de 0 y 1, generalmente implementada por mecanismos eléctricos. En general el lenguaje maquina es difícil de entender para los humanos por este motivo hacemos uso de lenguajes más parecidos a los lenguajes naturales.

Se denomina lenguaje máquina a la serie de datos que la parte física de la computadora o hardware, es capaz de interpretar. El lenguaje máquina fue el primero que empleo el hombre para la programación de las primeras computadoras.

Características

- El lenguaje máquina realiza un conjunto de operaciones predeterminadas llamadas micro operaciones.
- Las micro operaciones sólo realizan operaciones del tipo aritmética (+, -, *, /), lógicas (AND, OR, NOT) y de control (secuencial, decisión, repetitiva).
- El lenguaje máquina es dependiente del tipo de arquitectura. Así un programa máquina para una arquitectura Intel x86 no se ejecutará en una arquitectura Power PC de IBM (al menos de manera nativa).
- Algunos microprocesadores implementan más funcionalidades llamado CISC, pero son más lentos que los RISC ya que estos tienen registros más grandes.

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.

Ventajas

- Mayor adaptación al equipo.
- Máxima velocidad con mínimo uso de memoria.

Desventajas

- Imposibilidad de escribir código independiente de la máquina.
- Mayor dificultad en la programación y en la comprensión de los programas.
- El programador debe conocer más de un centenar de instrucciones.
- Es necesario conocer en detalle la arquitectura de la máquina

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.

La administración de la memoria es un proceso hoy en día muy importante, de tal modo que su mal o buen uso tiene una acción directa sobre el desempeño de memoria. En general un ensamblador tiene un administrador de memoria más limitado que un compilador; 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.

La memoria principal puede ser considerada como un arreglo lineal de localidades de almacenamiento de un byte de tamaño. Cada localidad de almacenamiento tiene asignada una dirección que la identifica

Se distinguen los siguientes propósitos del sistema de administración de memoria:

Protección.

Si varios programas comparten la memoria principal, se debería asegurar que el programa no sea capaz de cambiar las ubicaciones no pertenecientes a él. Aunque una acción de escritura puede tener efectos más graves que una de lectura, esta última tampoco debería estar permitida, para proporcionar algo de privacidad al programa.

Compartimiento.

Este objetivo parece contradecir al anterior, sin embargo a veces es necesario para los usuarios poder compartir y actualizar información (por ejemplo, en una base de datos) y, si se organiza la tarea de entrada a la misma, se puede evitar el tener varias copias de la rutina.

Reubicación.

La técnica de multiprogramación requiere que varios programas ocupen la memoria al mismo tiempo. Sin embargo no se sabe con anticipación donde será cargado cada programa por lo que no es práctico usar direccionamiento absoluto de memoria.

Organización física.

Debido al costo de una memoria principal rápida, éste se usa en conjunto con una memoria secundaria mucho más lenta (y por consiguiente, barata) a fines de extender su capacidad.

Organización lógica.

Aunque la mayor parte de las memorias son organizadas linealmente con un direccionamiento secuencial, esto difícilmente concuerda con el camino seguido por el programa, debido al uso de procedimientos, funciones, subrutinas, arreglos, etc.

Conclusiones

Los lenguajes de bajo nivel están muy apegados a la máquina, algunas empresas de antivirus y videojuegos, siguen programando pequeñas rutinas en lenguaje ensamblador por su rapidez y a pegamiento al computador. Esto es debido, a que un programa en ensamblador, es ejecutado casi directamente por el computador, ya que hablan casi el mismo idioma.

El lenguaje que la máquina que utiliza es el binario el que más usamos son los lenguajes naturales que son más sencillos para su comprensión el lenguaje de maquina tiene como características que hace operaciones aritméticas lógicas y de control, depende de la arquitectura de la máquina para su realización, el direccionamiento este donde se va ejecuta el programa en cual debe estar almacenado en la memoria principal puede ser directa o indirecta es cuando se inicia la ejecución del programa. Lenguaje ensamblador es un traductor de códigos de baja nivel está directamente en la máquina para su uso.

Conceptos

Generador: es un programa que toma como su entrada una especificación de la sintaxis de un lenguaje en alguna forma, y produce como su salida un procedimiento de análisis sintáctico para ese lenguaje

Código objeto: el código que resulta de la compilación del código fuente. Puede ser en lenguaje máquina o bytecode, y puede distribuirse en varios archivos que corresponden a cada código fuente compilado.

Optimizar: determinar los valores de las variables que intervienen en un proceso o sistema para que el resultado que se obtenga sea el mejor posible

Registros: son un tipo de dato estructurado formado por la unión de varios elementos bajo una misma estructura.

Ensamblador: se refiere a un tipo de programa informático que se encarga de traducir un fichero fuente escrito en un lenguaje ensamblador, a un fichero objeto que contiene código máquina, ejecutable directamente por el microprocesador.

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

Cargador: es un programa que realiza la función de carga, pero muchos cargadores también incluyen relocalización y ligado.

Macroinstrucción: se refiere a una instrucción en lenguaje ensamblador que es equivalente a otro grupo de instrucciones que, en conjunto, realizan una tarea más compleja.

Bifurcaciones: es la acción de separar algo en varias partes

Parámetros: son variables utilizadas para recibir valores de entrada en una rutina, subrutina o método.

Abstracción: es el proceso de separar ideas de instancias específicas de esas ideas en el trabajo

Multiprogramación: Es técnica por la que dos o más procesos pueden alojarse en la memoria principal y ser ejecutados concurrentemente por el procesador o CPU.

Bibliografía o Referencias

-Mark Francisco. (2016). Generación de código objeto. 28-Abr-2018, de acuario blogspot
Sitio web: <http://acaurio.blogspot.mx/2016/11/unidad-4-generacion-de-codigo-objeto.html>

-ITPN. (2014). Unidad IV: Generación de código objeto. 01-May-18, de INSTITUTO
TECNOLÓGICO DE PIEDRAS NEGRAS Sitio web:
<http://itpn.mx/recursosisc/7semestre/leguajesyautomatas2/Unidad%20IV.pdf>

-Orlando Medina Alonso. (2013). GENERACIÓN DE CÓDIGO OBJETO. 02-May-2018, de
academia.edu Sitio web:

http://www.academia.edu/10686636/GENERACION_DE_CODIGO_OBJETO

O

Reporte

Un generador de código objeto es el que transforma el código intermedio ya optimizado a un código objeto o de bajo nivel, lo que este hace es tomar el código intermedio y a partir de ahí genera el código para la máquina.

Los registros son la memoria principal de la computadora, hay varios tipos de registros que son utilizados para determinados tipos de funciones, los registros se clasifican en registros de datos que son de uso general y pueden ser usados como registros de 8 bits, los registros de segmentos que son los que dirigen las instrucciones del programa, registros punteros de pila, estos tienen la dirección relativa al segmento y se utilizan para fijar el puntero de pila y acceder a los elementos

El lenguaje Ensamblador es un tipo de lenguaje de bajo nivel que se usa para hacer programas en código máquina, las características de estos es que el programa sustituye los códigos por lo equivalente a código máquina, los programas que se hacen son portables de máquina a máquina, los lenguajes ensambladores se clasifican en básicos que son de muy bajo nivel y ofrecen nombres simbólicos a las instrucciones, y modulares que son descendientes de los básicos.

El lenguaje máquina proporciona la abstracción del microprocesador, solo lo puede entender una computadora se basan en una lógica binaria, el lenguaje máquina es difícil de entender por las personas y es por esto que usamos lenguajes más parecidos a los naturales, la principal característica de estos es que realizan un conjunto de operaciones predeterminadas, solo realizan operaciones de tipo aritmética y de control, las ventajas es que se adaptan más al equipo, son más rápidos y usan menos memoria, las desventajas son que no se puede escribir código independiente de la máquina, son más difíciles de hacer y de comprender y se debe conocer la arquitectura de la máquina para la que se va a hacer.

La administración de memoria es un proceso muy importante y su uso ya sea bueno o malo repercute sobre el desempeño de la memoria, los ensambladores tienen memoria más limitada de los compiladores, la memoria principal se considera como un arreglo lineal de localidades las cuales almacenan un byte de tamaño.