

Instituto Tecnológico de Nuevo Leon



Lenguaje y Autómatas II

Unidad 4: Generación de código objeto

Nombre: Erick Mauricio Medina Hernandez

Matricula: #13480549

Maestro: Ing. Juan Pablo Rosas Baldazo.

Proyecto 4: Elaboración de un resumen que incluya cada uno de los subtemas de la unidad.

Guadalupe N.L 27 de noviembre del 2018

Índice

Introducción.....	3
Capítulo 1: Registros.....	4
Capítulo 2: Lenguaje Ensamblador.....	5
Capítulo 3: Lenguaje Maquina.....	6
Capítulo 4: Administración de Memoria...	7
Reporte	8
Conceptos	9
Bibliografía o Referencias.....	10

Introducción

En este trabajo se hablará sobre los capítulos que se lleva a cabo en esta unidad, ¿Qué son los registros? y ¿para que nos pueden ser de utilidad?, además se verá sobre lo que es el lenguaje ensamblador, lenguaje máquina se mencionara un poco sobre en qué nos podrá servir y para qué es útil la administración de memoria.

PROYECTO 4 GENERACIÓN DE CÓDIGO OBJETO

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. Debido a la complejidad del software actual se necesitan de asistentes para poder instalar y ejecutar un programa. Operar sobre registros es más rápido y eficiente que operar sobre memoria. Por ello, la adjudicación eficiente de registros tiene un gran impacto en la performance.

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 (assignation), se elige el registro específico para cada variable.

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. También se usa en las operaciones aritméticas.

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. Una macroinstrucción es el equivalente a una función en un lenguaje de alto nivel.

Una ventaja importante del uso de ensamblador, es que se encarga de administrar de manera transparente para el usuario la creación de memoria, las bifurcaciones y el paso de parámetros y nos permite acceder directamente a los recursos de la máquina para un mejor desempeño.

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.

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.

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. 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. El direccionamiento relativo hace a los programas relocizables e independientes. Si la dirección base es el inicio de la memoria fija el direccionamiento pasa a ser una variante de direccionamiento absoluto.

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. 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, ésta 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.

Reporte

En cada punto que se explica en el tema de generación de código objeto. Se habla en cada capítulo de lo importante que son los registros y los lenguajes tanto ensamblador como máquina y por último como se administra una memoria. En el primer tema de los registros, son muy importantes ya que son la memoria principal de la computadora, ya que con estos la CPU solamente puede procesar datos que se encuentren en registros, existen varios registros de propósito general y algunos de uso exclusivo. El siguiente tema que es el lenguaje de ensamblador nos explica que es un traductor de código de bajo nivel a un código, ejecutable directamente por la máquina para la que se ha generado. Una de las características que tiene es que Cuando el programa lee un archivo escrito en lenguaje ensamblador y sustituye cada uno de los códigos mnemotécnicos por su equivalente código máquina. El otro tema que es el lenguaje máquina nos da una explicación de entendible por las computadoras, se basa en una lógica binaria de 0 y 1, generalmente implementada por mecanismos eléctricos. El lenguaje máquina es difícil de entender para los humanos, por este motivo hacemos uso de lenguajes más parecidos a los lenguajes naturales. Y una característica que tiene es que Realiza un conjunto de operaciones predeterminadas llamadas micro operaciones. Y por último la administración de memoria y esta trata de 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.

Conceptos

Compartición: Permitir a procesos diferentes acceder a la misma porción de memoria.

Organización Física: Permitir el intercambio de datos en la memoria primaria y secundaria.

Organización Lógica: Permitir que los programas se escriban compilables y ejecutables.

Protección: Evitar el acceso a posiciones de memoria sin el permiso expreso.

Registros de segmento: Se utiliza para alinear en un límite de párrafo y su dirección en un registro.

Registros de uso general: acumulador, de base, contador, y de datos.

Reubicación: Permitir el recalcu de direcciones de memoria de un proceso reubicado.

Mnemotécnicos: sirve para ayudar a la memoria a retener una cosa.

Bytecode: es un código intermedio más abstracto que el código máquina.

Distribución: es el proceso en el que el programa generado puede ejecutarse en otras máquinas.

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

Bibliografía o Referencias

Felipe Escobar. (2014). Generacion de codigo objeto. 2014, de Mktex Sitio web:
<http://itpn.mx/recursosisc/7semestre/leguajesyautomatas2/Unidad%20IV.pdf>

**Mark Francisco, Noviembre 2016, Generador de Código Objeto,
<http://acaurio.blogspot.mx/2016/11/unidad-4-generacion-de-codigo-objeto.html>**