INSTITUTO TECNOLÓGICO DE NUEVO LEÓN

Ingeniería En Sistemas Computacionales

Lenguajes y Autómatas II

U4

Tema: Generación de código objeto

Proyecto 4 Resumen

Profesor.
Juan Pablo Rosas Baldazo

Presenta.

Carlos Enrique Bernal Araujo

15480004

Índice

Introducción

En este documento se verá lo que son los registros y para qué nos pueden ser de utilidad, se hablara sobre lo que es lenguaje ensamblador, el lenguaje máquina y se menciona un poco sobre en qué nos resulta útil la administración de memoria.

1. Registros

Los registros son la memoria principal de la computadora, son los elementos más valiosos y escasos en la fase de generación de código, ya que el CPU solamente puede procesar datos que se encuentren en registros, existen varios registros de propósito general y algunos de uso exclusivo.

Algunos registros de propósito general son utilizados para cierto tipo de funciones, existen acumuladores, puntero de instrucción, de pila, etc.

La UCP o CPU tiene 14 registros internos, cada uno de ellos de 16 bits, los bits están enumerados de derecha a izquierda, de tal modo que el bit menos significativo es el bit 0.

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.

Estos registros son de uso general y también pueden ser utilizados como registros de 8 bits, para utilizarlos como tales es necesario referirse a ellos como, por ejemplo: AH y AL, que son los bytes alto (high) y bajo (low) del registro AX. Esta nomenclatura es aplicable también a los registros BX, CX y DX.

Además, las instrucciones que implican operandos en registros son más cortas y rápidas que las de operandos en memoria. El uso de registros se puede dividir en dos sub problemas:

- Durante la asignación de registros: aquí se selecciona el conjunto de variables y/o constantes que residirán en los registros en un momento del programa.
- Durante una fase posterior a la anterior: en esta se escoge el registro especifico en el que residirá una variable.

2. Lenguaje ensamblador

El ensamblador es un traductor de código de bajo nivel a un código, ejecutable directamente por la máquina para la que se ha generado. Fue la primera a abstracción de un lenguaje de programación, posteriormente aparecieron los compiladores.

Algunas de sus características pueden ser:

- Cuando 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.
- El programa escrito en lenguaje ensamblador se denomina código fuente (*asm)
- El programa ensamblador proporciona a partir de este fichero el correspondiente código máquina que suele tener la extensión (*.hex)

Se puede clasificar en 2 tipos:

- Ensambladores básicos: estos 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 también llamados macro ensambladores: Descendientes de los ensambladores básicos, fueron populares en las décadas de los 50 y los 60, fueron antes de la generalización de los lenguajes de alto nivel, un 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.

3. Lenguaje máquina

El lenguaje máquina solo es 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.

Sus características pueden ser:

- Realiza un conjunto de operaciones predeterminadas llamadas micro operaciones.

- Las micro operaciones solo 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.

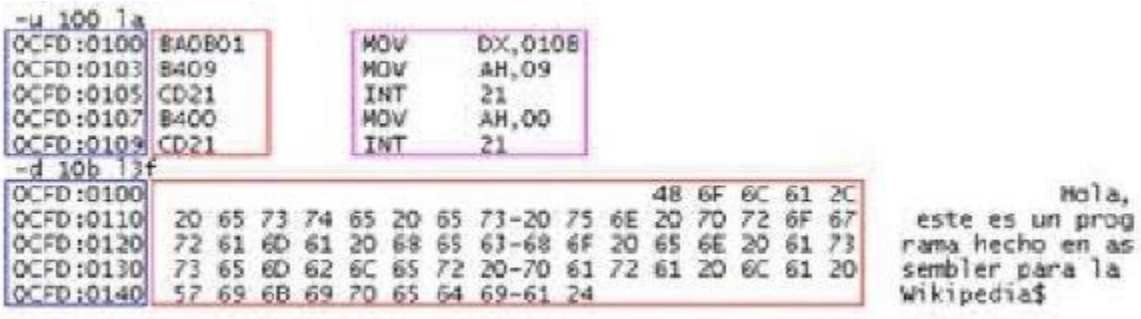
En programación, se le llama código objeto al código que resulta de la compilación del código fuente. Que consiste en lenguaje máquina o bytecode y se distribuye en varios archivos que corresponden a cada código fuente compilado.

En general se usa este tipo de lenguaje para programar controladores (drivers).

Algunas ventajas o inconvenientes

- Mayor adaptación al equipo
- Posibilidad de obtener la máxima velocidad con un mínimo uso de memoria
- Mayor dificultad en la programación y en la comprensión de los programas
- El programador debe conocer más de un centenario de instrucciones
- Es necesario conocer en detalle la arquitectura de la máquina

Ejemplo: lenguaje máquina del Intel 8088



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 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 pertenecientica 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 concuerde con el camino seguido por el programa, debido al uso de procedimientos, funciones, subrutinas, arreglos, etc

Conclusión:

Al realizar este documento noté que se nos va explicando cómo se manejan los espacios de memoria en la computadora y el cómo puede llegar a ser trabajar con el lenguaje ensamblador y el lenguaje máquina, pero para este último a lo que leí es más tedioso porque el programador debe saber un centenar de instrucciones.

Conceptos

- Mnemotécnicos: sirve para ayudar a la memoria a retener una cosa
- Bytecode: es un código intermedio más abstracto que el código máquina

Bibliografía

S.A, S.F, Unidad IV: Generación de código objeto, http://itpn.mx/recursosisc/7semestre/leguajesyautomatas2/Unidad%20IV.pdf

S.A, S.F, Un. VIII. Generación de Código Objeto, https://ingarely.files.wordpress.com/2012/11/unidad-viii.pdf

Orlando Medina Alonso, S.F, GENERACIÓN DE CÓDIGO OBJETO, http://www.academia.edu/10686636/GENERACI%C3%93N_DE_C%C3%93DIGO_OBJETO

Valeria Dimas Ortega, 30 noviembre 2016, Unidad 4 Generación de Código Objeto, http://unidad4generacion.blogspot.mx/

Jiovanny Jiménez Moreno, 20 de mayo del 2014, Generación de Código Objeto, https://prezi.com/tphqsdnxonz7/generacion-de-codigo-objeto/

S.A, S.F, 1.3.7. Generador de Código Objeto, http://cidecame.uaeh.edu.mx/lcc/mapa/PROYECTO/libro32/autocontenido/autocon/137_generador_de__cdigo_objeto.html

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

Reporte

Bueno para comenzar los registros son lo principal de una computadora, porque el CPU solo trabaja con lo que está dentro de los registros, el CPU tiene 14 registros internos y cada uno tiene 16 bits.

Esos registros se pueden clasificar en:

- Registros de datos
- Registros de segmentos
- Registros de punteros de pila
- Registros de índices
- Puntero de instrucciones
- Registro de banderas

El lenguaje ensamblador sirve como traductor de código de bajo nivel, se podría decir que fue de los primeros en salir de un lenguaje de programación, luego aparecieron los compiladores, este puede leer un archivo escrito en lenguaje ensamblador y sustituye cada uno de los códigos que los retienen por su equivalente código máquina.

El lenguaje maquina pues como su nombre lo dice solo lo entienden las computadoras, está basado en una lógica binaria de 0 y 1.

Es característico por realizar un conjunto de operaciones predeterminadas llamadas micro operaciones que vienen siendo aritméticas tales como suma, resta, multiplicación y división; pero también algunas lógicas como AND, OR y NOT, y creo que hasta ciclos también.

La administración de memoria trata de saber la posición de memoria en la que los diferentes símbolos del programa almacenan la información, es un proceso muy importante. Cada localidad de almacenamiento tiene asignada una dirección que la identifica.

Hay varios propósitos del sistema para la administración de memoria y son los siguientes:

- Protección
- Compartimiento
- Reubicación
- Organización Física
- Organización lógica