INSTITUTO TECNOLÓGICO DE NUEVO LEÓN

Lenguajes y Autómatas II

UNIDAD # 4

Nombre de la Unidad:

Generación de código Objeto

Catedrático: Juan Pablo Rosas

Nombre: Alondra Sarahi Sanchez Saucedo

Matrícula: 14480840

Guadalupe N.L. a 03 de 05 del 2018

Introducción

En el siguiente contenido mostraremos alguna información acerca de los lenguajes de programación de alto y bajo nivel como lenguaje ensamblador y lenguaje maquina también se mostrar información acerca de los registros en la computadora y sus tipos con los que cuenta y uno de los temas muy importante que es la administración de la memoria, se muestra un resumen de los diferentes temas ya mencionados para analizar y entender mejor su definición.

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.

4.1 REGISTROS

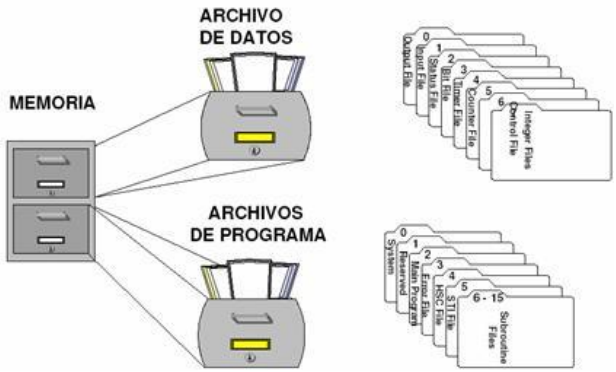
¿Qué son?

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.

Los registros son espacios físicos dentro del microprocesador con capacidad de 4 bits hasta 64 bits dependiendo del microprocesador que se emplee.

¿Quiénes lo utilizan?

Antes de nada, para el desarrollo de esta parte hablaremos indistintamente de registros de activación o de marcos de pila. Esto se debe a que en la documentación encontrada sobre el manejo de los registros ebp y esp se hace mención a dicho concepto de marco de pila. Puesto que el lenguaje permite recursividad, los registros de activación se asignan dinámica mente.



Distribución

La UCP o CPU tiene 14 registros internos, cada uno de ellos de 16 bits (una palabra). 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.

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.

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.

¿Cuál es su aplicación en la generación de códigos?

1. usar el registro de y si está en un registro que no tiene otra variable, y además y no está viva ni tiene uso posterior. Si no:
2. usar un registro vacío si hay. Si no:
3. usar un registro ocupado si op requiere que x esté en un registro o si x tiene uso Posterior. Actualizar el descriptor de registro. Si no:
4. usar la posición de memoria de x

4.2 Lenguaje ensamblador

¿Qué es?

El lenguaje Assembly (Urbina, 2011) (a veces mal llamado "Ensamblador" por su traducción literal al español) es un tipo de lenguaje de bajo nivel utilizado para escribir programas informáticos, y constituye la representación más directa del código máquina específico para cada arquitectura de computadora

Segunda generación de lenguajes

Versión simbólica de los lenguajes máquina (Urbina, 2011) (MOV, ADD).La comunicación en lenguaje de máquina es particular de cada procesador que se usa, y programar en este lenguaje es muy difícil y tedioso, por lo que se empezó a buscar mejores medios de comunicación con ésta. Los lenguajes ensambladores tienen ventajas sobre los lenguajes de máquina.

Este lenguaje fue usado ampliamente en el pasado para el desarrollo de software, pero actualmente sólo se utiliza encontradas ocasiones, especialmente cuando se requiere la manipulación directa del hardware o se pretenden rendimientos inusuales de los equipos

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.

Clasificación:

- **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, fueron antes de la generalización de los lenguajes de alto nivel. Un macro instrucción es el equivalente a una función en un lenguaje de alto nivel.

Operaciones básicas

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 para el cual programas.
Pero la idea básica es:

--definir que 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.).

4.3 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. Una instrucción en lenguaje máquina puede representarse de la siguiente forma: 01101100101001001111011O. Esta secuencia es fácilmente ejecutada por la computadora, pero es de difícil interpretación, siendo aún más difícil la interpretación de un programa (conjunto de instrucciones) escrito de esta forma.

Esta dificultad hace que los errores sean frecuentes y la corrección de los mismos costosa, cuando no imposible, al igual que la verificación y modificación de los programas.

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, de control y repetitiva). El lenguaje maquina es dependiente del tipo de arquitectura. Así un programa máquina para una arquitectura Intel x86 no sé ejecutara en una arquitectura Power PC de IBM (al menos de manera nativa).

Algunos microprocesadores implementan mas funcionalidades llamado CISC, pero son más lentos que los RISC ya que estos tienen registros más grandes.

Ventajas

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

Desventajas

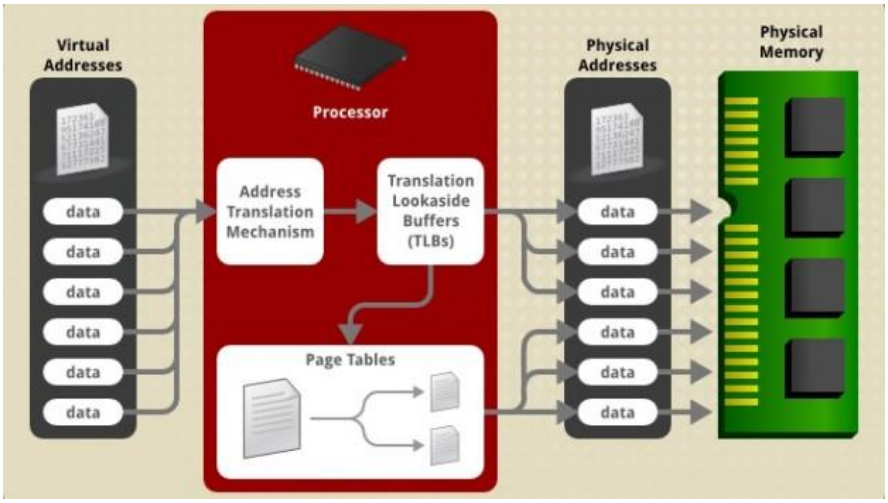
- Imposibilidad de escribir código independiente de la máquina.
- Mayor dificultad en la programación y en la comprensión de los programas.
- El programador debe conocer más de un centenar de instrucciones.
- Es necesario conocer en detalle la arquitectura de la máquina.

4.4 Administrador de memoria

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 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

En esta unidad habla acerca de generador de cogido objeto y nos muestra algunos lenguajes como el lenguaje ensamblador este más que nada es un traductor de bajo nivel este era uno de los más usados anterior mente para crear los software pero con el tiempo fueron saliendo mas lenguajes que facilitan mejor su forma de entenderlos este lenguaje cuenta con dos tipos ya sea ensamblador básico o ensamblador modulares en el ensamblador básico hacia referencia a traducir los códigos con simbologías estos eran de muy bajo nivel mientras que los ensambladores modulares hacían referencia mas a un lenguaje de alto nivel este te permitía tener un control mas avanzado en estructuras, otro de los lenguajes de los cual nos mostraba era el lenguaje maquina este caracteriza por ser un lenguaje difícil de entender por los humanos ya que su conformación fue para las computadoras y este se caracteriza por tener 0 y 1 esto no quiere decir que lo podremos traducir solo que es un poco tedioso estar traduciéndolo este lenguaje se caracteriza también por realizar operaciones de tipo aritmética ya sea la suma, división, multiplicación o resta y los lógicas que son AND, OR, NOT en materias anteriores ya hemos realizado algunos ejemplos con este lenguaje maquina y es divertido estar traduciendo algún mensaje que quieras descifrar ya que como lo dije se caracteriza por tener mensaje como 0 y 1 por ejemplo para realizar una palabra tan corta teníamos que descifrar una cantidad tan grandes de estos numero como la palabra (Hola) lo saque de una página su traducción y hace referencia a esta cantidad 01101000 01101111 01101100 01100001 es tan impresionante la cantidad de valores que debemos de utilizar para traducir una palabra tan corta, los registros son las instrucciones que se guardan en la memoria, distribución en que se pueden ejecutar o usar en otras maquinas hay diferentes tipos de registros como de índices, de segmentos, de pilas etc. Estos registros son espacios dentro del microprocesador ya dependiendo que microprocesador tengas para su capacidad, el administrador de memoria este es un tema importante ya que debes de saber con que capacidad de memoria cuentas es decir si tienes poca memoria y tienes un programa que abarca toda esa memoria quizás tu computador este un poco lento debes tener en cuenta como esta distribuida o como la vas a utilizar ya que muchas veces tienes muy poca capacidad y utilizas demasiada y eso te genera que no tengas un buen rendimiento por eso es recomendable contar con algún software que te permita visualizar o analizar el rendimiento de tu memoria y como la están utilizando los diferentes programas que estés utilizando

Conceptos

- registros acumuladores: son almacenados temporalmente los resultados aritméticos y lógicos intermedios que serán tratados por el circuito operacional de la unidad aritmético-lógica (ALU)

- puntero de instrucción: es un registro del procesador de un computador que indica la posición donde está el procesador en su secuencia de instrucciones

- recursividad: Es una técnica utilizada en programación que nos permite que un bloque de instrucciones se ejecute un cierto número de veces (el que nosotros determinemos).

- registros de activación: almacena la información sobre las subrutinas activas de un programa de computadora.

- arquitectura Power PC de IBM: es el nombre original de la arquitectura de computadoras de tipo RISC, que fue desarrollada por IBM, Motorola, y Apple.

- assembler: es un lenguaje de programación de bajo nivel. Consiste en un conjunto de mnemónicos que representan instrucciones básicas para los computadores, microprocesadores, microcontroladores y otros circuitos integrados programable

- CISC: (complex instruction set computer) Computadoras con un conjunto de instrucciones complejo.

- Risc: (reduced instruction set computer) Computadoras con un conjunto de instrucciones reducido

- direccionamiento absoluto de memoria: El campo de operando contiene una dirección en memoria, en la que se encuentra la instrucción. Y no se cancela.

- direccionamiento secuencial: La ejecución secuencial no está considerada un modo de direccionamiento en algunos computadores. La mayoría de instrucciones en la mayoría de las arquitecturas de CPU son instrucciones secuenciales. Debido a que la mayoría de las instrucciones son de tipo secuencial, los diseñadores de la CPU a menudo añaden características que deliberadamente sacrifican el rendimiento, por un lado, y por otro las instrucciones de la rama de instrucciones a fin de que estas instrucciones secuenciales corran más rápido.

- mnemónicas: un mnemónico o nemónico es una palabra que sustituye a un código de operación (lenguaje de máquina), con lo cual resulta más fácil la programación, es de aquí de donde se aplica el concepto de lenguaje ensamblador.

- compilador: es un programa informático que traduce un programa que ha sido escrito en un lenguaje de programación a un lenguaje común, reúne diversos elementos o fragmentos en una misma unidad. 1 usualmente lenguaje de máquina, aunque también puede ser traducido a un código intermedio (bytecode) o a texto.

- punteros: un puntero es un objeto del lenguaje de programación, cuyo valor se refiere a (o "apunta a") otro valor almacenado en otra parte de la memoria del ordenador utilizando su dirección.

Bibliografía

http://acaurio.blogspot.mx/2016/11/unidad-4-generacion-de-codigo-objeto.html

Francisco ,M.. (noviembre 29,2016). unidad 4 Generación de código objeto. marzo 03,2018, de Blogger Sitio web: http://acaurio.blogspot.mx/2016/11/unidad-4-generacion-de-codigo-objeto.html