

Instituto tecnológico de nuevo león.

Ingeniería en sistemas computacionales.



Materia: Lenguajes y Autómatas II

Tarea: Resumen de la unidad IV

Alumno: Gaspar Cruz Melendres.

Numero Control: 14480839

Catedrático: Juan Pablo Rosas Baldazo

Guadalupe, Nuevo León. 4 de mayo del 2017

1. Introducción

En este trabajo estaremos viendo lo que es generación de códigos objetos que se trata de ir viendo lo que son los registro, la forma de lenguaje ensamblador, el lenguaje máquina y la administración de memoria, las cuales son fundamentales para identificar los conceptos de los códigos que se comprende desde el usuario asta con la máquina.

Por otro lado, esto se enfoca a la transformación de código intermedio para la optimización en los códigos objetos de bajo nivel, siendo que con esto podremos agilizar y tener eficacia en lo que es la forma de operación de los códigos, y no tener tantas redundancia o líneas innecesarias en el código en sí mismo ya que con ello podremos ahorrar códigos para su menor tiempo de respuesta en la ora de ejecución.

2. Capítulo 1: Registros

¿Qué son?

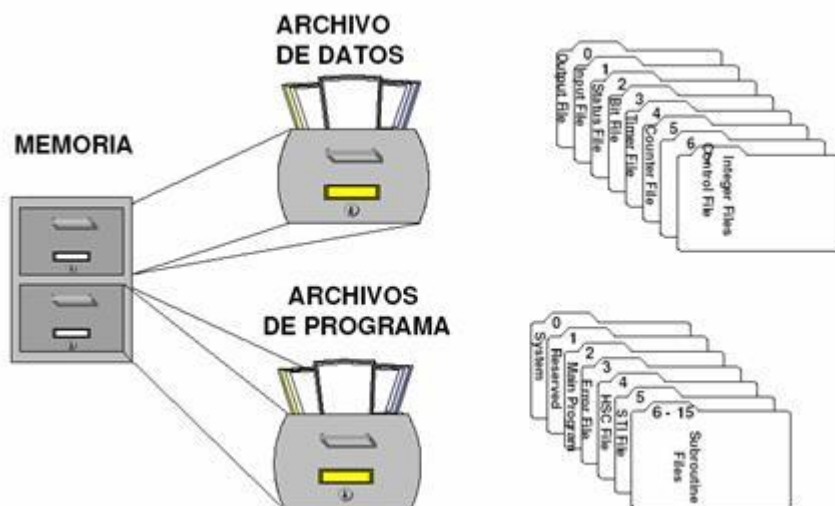
Los registros son la memoria principal de la computadora. Existen diversos registros de propósito general y otros de uso exclusivo. Algunos registros de propósito general son utilizados para cierto tipo de funciones. Existen registros acumuladores, puntero de instrucción, de pila, etc.

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

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.

¿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 dinámicamente.







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.

¿Cuáles 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

3. Capítulo 2: Lenguaje ensamblador

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. Una macroinstrucción 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. 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.

Resultado de imagen para Lenguaje MáquinaSe 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: 011011001010010011110110. 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 se ejecutara 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.

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.

5. Capítulo 4: Administración de memoria

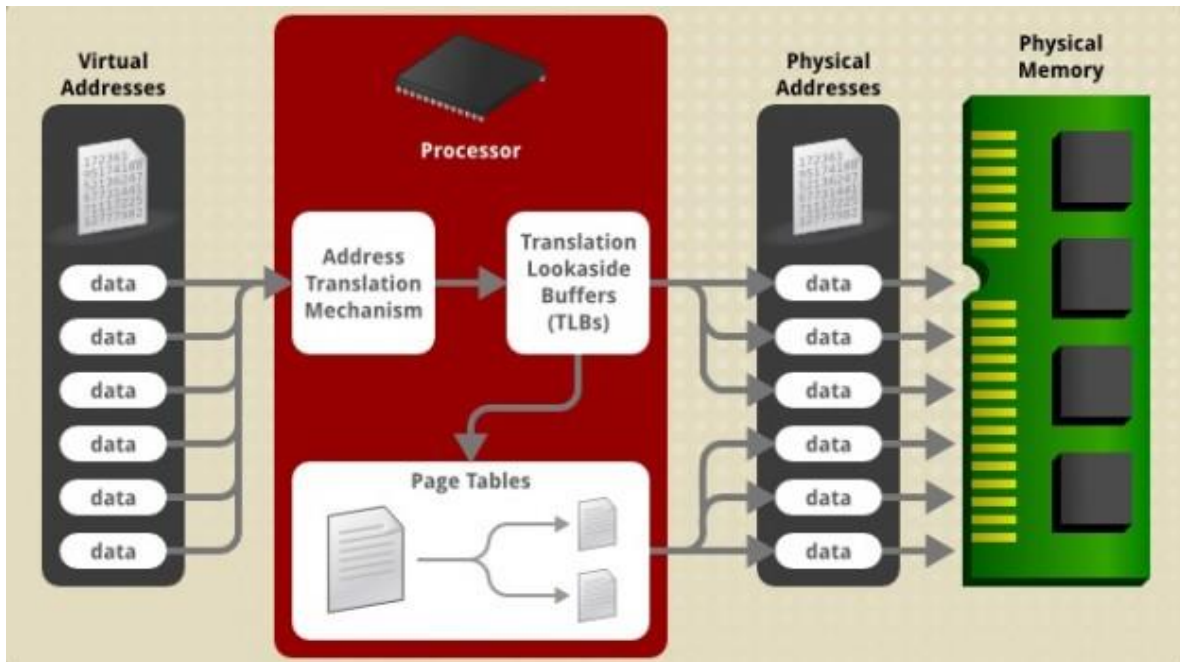
La administración de la memoria es un proceso hoy en día muy importante, de tal modo que su mal o buen uso tiene una acción directa sobre el desempeño de memoria. En general un ensamblador tiene un administrador de memoria más limitado que un compilador; en la mayoría de los lenguajes de programación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

6. Conclusiones

Como podemos ver en esta investigación que realice nos damos cuentas que los registros son muy valiosos ya que con ellos aplica en la fase de gestión de códigos ya que por alguna parte también son escasos en ese punto ya que en el CPU debe de tener información de los registros para poder procesar o administrar los datos que él requiera.

No solo podemos utilizar los registros como acumuladores ya que con ello podemos utilizar lo que son las pilas ya que es otra forma de administrar la información que se le da a esas faces.

Otra de las investigaciones fue la comparación de lenguaje maquina con el lenguaje ensamblador, ya que en el lenguaje máquina solo entiende lo que son las computadoras ya que habla de forma binaria de 0 y 1 siendo que eso las personas o usuarios que utilizan las computadoras no lo comprende, sin en cambio ellas es su lenguaje natal.

Mientras el lenguaje ensamblador es un lenguaje de bajo nivel que se asemeja en la forma de lenguaje maquina pero que es entendible para el usuario, para su comprensión de este lenguaje debemos de entender bien el concepto.

En el caso de administración de memoria esto anos ayudara a la forma correcta de limpiarla o de administrarla ya que al momento de estar checando cosas puede afectar a la memoria llenarse de información que no lo ocupa el usuario siendo que por ello debemos de tener un control o una forma de verificar el almacenamiento de ella, viendo por otro lado monitorear la memoria a su cien por ciento ya que al veces también puede haber programa los cuales tendrán la memoria llenada y tendrá una baja rendición de la memoria y se tendrá que hacer mantenimientos correctivos a la memoria.

En esta investigación también se ve lo que son los propósitos de administración de la memoria los cuales son muy complejos ya que con estos puntos tendrás un mejor rendimiento en la operatividad en donde se requiera la memoria.

7. Conceptos

- Bits: Un bit es un dígito del sistema de numeración binario. La capacidad de almacenamiento de una memoria digital también se mide en bits, pues esta palabra tiene varias acepciones.
- BYTE: Un byte está formado por 8 bits. Su valor puede oscilar entre 0-255. Es un sistema en base 2. Nosotros para facilitar la lectura de los números binarios, usaremos el sistema hexadecimal (sistema en base 16) por la rapidez y facilidad para leer.
- Pila: es un modelo matemático de un sistema que recibe una cadena constituida por símbolos de un alfabeto y determina si esa cadena pertenece al lenguaje que el autómata reconoce.
- Marcos de pila: Es una abstracción de una pila que proporciona el contexto de ejecución de un subproceso.
- Micro operaciones: Las micro operaciones que se encuentran con mayor frecuencia en las computadoras digitales se clasifican en cuatro categorías:
 - Micro operaciones de transferencia de registros
 - Micro operaciones aritméticas
 - Micro operaciones lógicas
 - Micro operaciones de corrimiento
- EBP: (Extended Base Pointer) Registro no volátil con dos usos comunes según el compilador que utilicemos, así puede desempeñar el papel de un registro como otro cualquiera o ser el puntero al marco de pila.
- ESP: (Extended Stack Pointer) Almacena un puntero a la parte inferior de la pila. Tras ejecutar una función el valor que tenía el registro al principio debe de coincidir con el asociado tras la función
- Multiprogramación: Se denomina multiprogramación a una técnica por la que dos o más procesos pueden alojarse en la memoria principal y ser ejecutados concurrentemente por el procesador o CPU.
- RISC: En arquitectura computacional, RISC (del inglés Reduced Instruction Set Computer, en español Computador con Conjunto de Instrucciones Reducidas) es un tipo de diseño de CPU generalmente utilizado en microprocesadores o microcontroladores con las siguientes características fundamentales:
 - Instrucciones de tamaño fijo y presentadas en un reducido número de formatos.
 - Sólo las instrucciones de carga y almacenamiento acceden a la memoria de datos

8. Bibliografía o Referencias

Mark, F.. (2016). *Generador de Código Objeto unidad 4*. Mayo 03, 2018, de Blogger. Sitio web: <http://acaurio.blogspot.mx/2016/11/unidad-4-generacion-de-codigo-objeto.html>

Martinez, J.. (2013). *Unidad IV*. Mayo 3, 2018, de Google Sitio web: <http://itpn.mx/recursosisc/7semestre/leguajesyautomatas2/Unidad%20IV.pdf>

Garcia, S.. (2013). *autómatas (registro)*. Mayo 03, 2018, de SlideShare Sitio web: https://es.slideshare.net/sam_server/expo-automatas