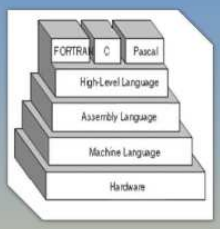
Introduccion

- El ordenador solo entiende el lenguaje de código binario o código maquina ,solo utiliza 0 y 1 para de codificar cualquier acción .

Lenguaje de bajo nivel

- Son lenguajes totalmente dependientes de la maquina .
- Dentro de este grupo se encuentra el lenguaje ensamblador.



Lenguaje Ensamblador

- Derivado del lenguaje maquina , formado por abreviaturas de letras y números llamadas mnemotécnicos .

¿Qué es un lenguaje ensamblador?

•Es un lenguaje en el que cada enunciado produce exactamente una instrucción maquina.

•Los lenguajes ensambladores tienen acceso a todas las características e instrucciones disponibles en la maquina.

•En resumen todo lo que puede hacerse en lenguaje maquina puede hacerse en lenguaje ensamblador.

Importancia

- El lenguaje ensamblador es importante por que el es considerado de primera generación a partir de el se derivaron todos los demás lenguajes hasta llegar a los de alto nivel.

Características

•Ensamblador es directamente traducible al Lenguaje de Máquina, y viceversa.

•La computadora no entiende directamente al Lenguaje Ensamblador; es necesario traducirle a Lenguaje de Máquina.

• Se utilizan traductores que convierten el código fuente (en Lenguaje Ensamblador) a código objeto.

•El usar los traductores de codigo son con el fin de facilitar la programacion y tener el control del harware.

Ventajas y desventajas del Lenguaje Ensamblador vs lenguaje de alto nivel

Lenguaje Ensamblador.

1. Velocidad
2. Eficiencia de tamaño.
3. Flexibilidad

Lenguaje de alto nivel.

1. Tiempo de programación
2. Programas fuente grandes
3. Peligro de afectar recursos inesperadamente.
4. Falta de portabilidad

Velocidad

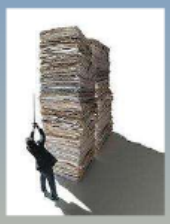
- Implica un proceso de cómputo adicional al que el programador quiere realizar.
- Un intérprete es siempre más lento que realizar la misma acción en Lenguaje Ensamblador.
- Los compiladores son mucho más rápidos que los intérpretes, pues hacen la traducción una vez y dejan el código objeto.
- Mayor parte de las veces, el código generado por un compilador es menos eficiente que el código equivalente que un programador escribiría.

Tamaño

- Existen programas donde el uso de las memoria es critico para esos casos es eficiente el lenguaje ensamblador por la mínima cantidad de recursos de los que dispone

Programa fuentes grandes

- Crecen los programas fuentes; simplemente, requerimos más instrucciones primitivas para describir procesos equivalentes. Esto es una desventaja porque dificulta el mantenimiento de los programas, y nuevamente reduce la productividad de los programadores.



Tiempo de programación

Requiere más instrucciones para realizar el mismo proceso.
Por otro lado, requiere de más cuidado por parte del programador

El proceso de ensamblado

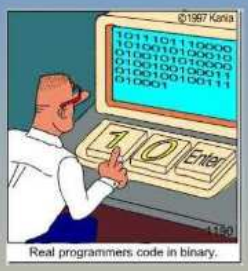
- El lenguaje interactúa directamente con los dispositivos de hardware y dispositivos lógicos como las memorias y el CPU

Instrucciones en ensamblador

- Mov.-mueve el valor de un registro o un numero hacia otro registro ejmplo mov Bx,5 , movAX,bx.
- add.-Sumar el valor de un registro a otro registro ADD BX,5
- sub.-Rest el valor de un registro o valor especifico a un registro sub cx,2
- inc incrementa en 1 el valor del registro inc bx
- dec.-Decrementa en 1 el valor del registo dec bx

Aplicaciones

- El uso del lenguaje ensamblador no es para la gente común y corriente, sino para profesionistas en el área de computación que están obligados a conocer este lenguaje, ya que proporciona una serie de características que no se pueden encontrar en los lenguajes de alto nivel.



Real programmers code in binary.

Aplicaciones

- Se puede acceder a cualquier localidad de la memoria RAM .
- Se pueden programar virus, debido a que se tiene un acceso total a casi todo el hardware de la computadora vía interrupciones de software
- Se pueden programar drivers de cualquier dispositivo.
- Programación de Microcontroloadores
- Creación de compiladores
- Se puede acceder directamente a los dispositivos de entrada y/o salida.

Ramas en las que se aplica

- Sistemas Embebidos: impresoras, cámaras, autos, juguetes, etc.
- Industria y Manufactura: adquisición datos y control, eg robots.
- Transporte y Aeronaútica: barcos, aviones, sondas espaciales, etc.
- Graficación, Multimedia, Cine y Video Juegos
- Procesamiento de Señales, Voz e Imágenes
- Armamento y Defensa

TIPOS DE ENSAMBLADORES

- ENSAMBLADORES CRUZADOS: Se denominan así a los ensambladores que se utilizan en una computadora que posee el procesador diferente al que tendrán las computadoras donde se va a ejecutar el programa objeto producido.

- ENSAMBLADORES RESIDENTES: Son aquellas que permanecen en la memoria principal de la computadora y cargar para su ejecución al programa objeto producido.

- MICRO ENSAMBLADORES: Al programa que indica al intérprete de instrucciones de la CPU como debe actuar se le denomina microprograma. El programa que ayuda a realizar este microprograma se llama micro ensamblador.

- MACRO ENSAMBLADORES: Son ensambladores que permiten el uso de macroinstrucciones.

- ENSAMBLADORES DE UNA FASE: leen una línea y la traducen directamente para producir una instrucción de lenguaje maquina o la ejecuta si se trata de una pseudosinstrucción. Se construye la tabla de símbolos a medida que aparecen las definiciones de variables, etiquetas, etc.

- ENSAMBLADORES DE DOS FASES: Realiza la traducción en dos etapas: 1° fase leen el programa fuente y construyen la tabla de símbolos, 2° fase vuelve a leer el programa fuente y pueden ir traduciendo totalmente pues reconocen la totalidad de los símbolos.

1. Importancia del lenguaje ensamblador

La importancia del lenguaje ensamblador radica principalmente que se trabaja directamente con el microprocesador; por lo cual se debe de conocer el funcionamiento interno de este, tiene la ventaja de que en el se puede realizar cualquier tipo de programas que en los lenguajes de alto nivel no lo pueden realizar. Otro punto sería que los programas en ensamblador ocupan menos espacio en memoria.

2. Ventajas y desventajas del Lenguaje Ensamblador

Ventajas

1. Velocidad .- Como trabaja directamente con el microprocesador al ejecutar un programa, pues como este lenguaje es el mas cercano a la máquina la computadora lo procesa mas rápido.
2. Eficiencia de tamaño.- Un programa en ensamblador no ocupa mucho espacio en memoria porque no tiene que cargan librerías y demás como son los lenguajes de alto nivel
3. Flexibilidad .- Es flexible porque todo lo que puede hacerse con una máquina, puede hacerse en el lenguaje ensamblador de esta máquina; los lenguajes de alto nivel tienen en una u otra forma limitantes para explotar al máximo los recursosde la máquina. O sea que en lenguaje ensamblador se pueden hacer tareas especificas que en un lenguaje de alto nivel no se pueden llevar acabo porque tienen ciertas limitantes que no se lo permite

Desventajas

Tiempo de programación .- Como es un lenguaje de bajo nivel requiere más instrucciones para realizar el mismo proceso, en comparación con un lenguaje de alto nivel. Por otro lado, requiere de más cuidado por parte del programador, pues es propenso a que los errores de lógica se reflejen más fuertemente en la ejecución.

Programas fuente grandes .- Por las mismas razones que aumenta el tiempo, crecen los programas fuentes; simplemente requerimos más instrucciones primitivas para describir procesos equivalentes. Esto es una desventaja porque dificulta el mantenimiento de los programas, y nuevamente reduce la productividad de los programadores.

Peligro de afectar recursos inesperadamente .- Que todo error que podamos cometer, o todo riesgo que podamos tener, podemos afectar los recursos de la maquina, programar en este lenguaje lo más común que pueda pasar es que la máquina se bloquee o se reinicialize. Porque con este lenguaje es perfectamente posible (y sencillo) realizar secuencias de instrucciones inválidas, que normalmente no aparecen al usar un lenguaje de alto nivel.

Falta de portabilidad.- Porque para cada máquina existe un lenguaje ensamblador; por ello, evidentemente no es una selección apropiada de lenguaje cuando deseamos codificar en una máquina y luego llevar los programas a otros sistemas operativos o modelos de computadoras.

3. Relación del lenguaje ensambladorcon los componentes internos del procesador

- En la memoria se almacena la información en celdas especiales llamados registros los cuales tienen un nivel alto y un nivel bajo.
- Unidad aritmética y lógica es la responsable de realizar como su nombre lo indica operaciones aritméticas y lógicas.
- Unidad de control Se encarga de coordinar de que los otros componentes ejecuten las

operaciones correctamente.

- Bus interno son los canales por donde pasa la información que la máquina va a procesar (bus de entrada) o procesada (bus de salida).

Registros de uso general
AX = Registro acumulador, dividido en AH y AL (8 bits cada uno).- Interviene en las operaciones aritméticas y lógicas, después de la operación arroja un resultado.
BX = Registro base, dividido en BH y BL.- Se utiliza en transferencias de datos entre la memoria y el procesador.
CX = Registro contador, dividido en CH y CL.- Se utiliza como contador en bucles(LOOP), en operaciones con cadenas(REP), y en desplazamientos(CL).
DX = Registro de datos, dividido en DH y DL.- Se utiliza en operaciones de multiplicación y división junto con Ax y en operaciones de entrada y salida de puertos, su mitad inferior DL contiene el número de puertos.

Registros de Estado
Hay nueve indicadores de un bit en este registro de 16 bits. Los cuatro bits más significativos están indefinidos, mientras que hay tres bits con valores determinados: los bits 5 y 3 siempre valen cero y el bit 1 siempre vale uno.
CF (Carry Flag, bit 0): Si vale 1, indica que hubo "arrastre" (en caso de suma) o "préstamo" (en caso de resta). Este indicador es usado por instrucciones que suman o restan números que ocupan varios bytes. Las instrucciones de rotación pueden aislar un bit de la memoria o de un registro poniéndolo en el CF.
PF (Parity Flag, bit 2): Si vale uno, el resultado tiene paridad par, es decir, un número par de bits a 1. Este indicador se puede utilizar para detectar errores en transmisiones.
AF (Auxiliary carry Flag, bit 4): Si vale 1, indica que hubo "arrastre" o "préstamo" del nibble (cuatro bits) menos significativo al nibble más significativo. Este indicador se usa con las instrucciones de ajuste decimal.
ZF (Zero Flag, bit 6): Si este indicador vale 1, el resultado de la operación es cero.
SF (Sign Flag, bit 7): Refleja el bit más significativo del resultado. Como los números negativos se representan en la notación de complemento a dos, este bit representa el signo: 0 si es positivo, 1 si es negativo.
TF (Trap Flag, bit 8): Si vale 1, el procesador está en modo paso a paso. En este modo, la CPU automáticamente genera una interrupción interna después de cada instrucción, permitiendo inspeccionar los resultados del programa a medida que se ejecuta instrucción por instrucción.
IF (Interrupt Flag, bit 9): Si vale 1, la CPU reconoce pedidos de interrupción externas. Si vale 0, no se reconocen tales interrupciones
DF (Direction Flag, bit 10): Si vale 1, las instrucciones con cadenas sufrirán "auto-decremento", esto es, se procesarán las cadenas desde las direcciones más altas de memoria hacia las más bajas. Si vale 0, habrá "auto-incremento", lo que quiere decir que las cadenas se procesarán de "izquierda a derecha".
OF (Overflow flag, bit 11): Si vale 1, hubo un desborde en una operación aritmética con signo, esto es, un dígito significativo se perdió debido a que tamaño del resultado es mayor que el tamaño del destino.

4. Relación entre el código binario y el lenguaje ensamblador

En el código binario se utilizan ceros y unos, mientras que el lenguaje ensamblador es una colección de símbolos mnemónicos que representan: operaciones, nombres simbólicos, operadores y símbolos especiales.

La relación entre estos dos lenguajes sería que el binario es el lenguaje que la máquina entiende y el ensamblador se acerca mas lenguaje de esta.

Manejo de la memoria: Direccionamiento (interno y externo)
El manejo de la memoria depende de que procesador tenga la máquina, entre los cuales a continuación se mencionan los siguientes:

- Memoria de Programa
- Memoria Externa de Datos
- Memoria Interna de Datos
- Registros de Funciones Especiales
- Memoria de Bit.

El espacio de la Memoria de Programa contiene todas las instrucciones, datos, tablas y cadenas de caracteres (strings) usadas en los programas. Esta memoria se direcciona principalmente usando el registro de 16 bits llamado Data Pointer. El tamaño máximo de la Memoria de Programa es de 64 Kbytes.
La Memoria Externa de Datos contiene todas las variables y estructuras de datos que no caben en la memoria interna del Microprocesador. Esta memoria se direcciona principalmente por el registro de 16 bits Data Pointer , aunque también se puede direccionar un banco de Memoria Externa de Datos de 256 bytes usando los dos primeros registros de propósito general .
El espacio de Memoria Interna de Datos funcionalmente es la memoria de datos más importante, ya que ahí es donde residen cuatro bancos de registros de propósito general; la pila o stack del programa; 128 bits de los 256 bits de un área de memoria direccionable por bit y todas las variables y estructuras de datos operadas directamente por el programa. El tamaño máximo de la Memoria Interna de Datos es de 256 bytes.
Contiene un espacio para los denominados Registros de Funciones Especiales destinado para los puertos de entrada/salida, temporizadores y puerto serie del circuito integrado. Estos registros incluyen al Stack Pointer; al registro de la palabra de estado del programa y al Acumulador. La cantidad máxima de Registros de Funciones Especiales es 128.
Todos los Registros de Funciones Especiales tienen direcciones mayores a 127 y se ubican en los 128 bytes superiores de la Memoria Interna de Datos. Estas dos áreas de la Memoria Interna de Datos se diferencian por el modo de direccionamiento usado para accesarlas. Los Registros de Funciones Especiales solo se pueden accesar usando el modo de direccionamiento Directo, mientras que los 128 bytes superiores solo se pueden accesar con el modo de direccionamiento Indirecto.
Por otra parte, el espacio de Memoria de Bit se usa para almacenar variables y banderas de un bit. El tamaño máximo de la Memoria de Bit es de 256 bits, 128 de los bits comparten su espacio con 16 bytes del espacio de la Memoria Interna de Datos y los otros 128 bits lo hacen con los Registros de Funciones Especiales.

La Escalabilidad

En telecomunicaciones y en ingeniería informática, es la propiedad deseable de un sistema, una red o un proceso, que indica su habilidad para extender el margen de operaciones sin perder calidad, o bien manejar el crecimiento continuo de trabajo de manera fluida.

El Microprocesador

Es un circuito integrado que contiene algunos o todos los elementos hardware, y el de CPU, que es un concepto lógico. Una CPU puede estar soportada por uno o varios microprocesadores, y un microprocesador puede soportar una o varias CPU. Un núcleo suele referirse a una porción del procesador que realiza todas las actividades de una CPU real.

Partes del Microprocesador

Unidad de Control. Todos los recursos de la computadora son administrados desde la unidad de control, cuya función es coordinar todas las actividades de la computadora.

Unidad aritmética-lógica (ALU). Cuando la unidad de control encuentra una instrucción que involucra aritmética o lógica, le pasa el control al segundo componente de la CPU.

El **bus de direcciones** es un canal del microprocesador totalmente independiente del bus de datos donde se establece la dirección de memoria del dato en tránsito.

El bus de dirección consiste en el conjunto de líneas eléctricas necesarias para establecer una dirección.La capacidad de la memoria que se puede direccionar depende de la cantidad de bits que conforman el bus de direcciones, siendo 2^n (dos elevado a la ene) el tamaño máximo en bytes del banco de memoria que se podrá direccionar con n líneas. Por ejemplo, para direccionar una memoria de 256 bytes, son necesarias al menos 8 líneas, pues 2^8 = 256. Adicionalmente pueden ser necesarias líneas de control para señalar cuando la dirección está disponible en el bus. Esto depende del diseño del propio bus.

El **bus de control** gobierna el uso y acceso a las líneas de datos y de direcciones. Como éstas líneas están compartidas por todos los componentes, tiene que proveerse de determinados mecanismos que controlen su utilización. Las señales de control transmiten tanto órdenes como información de temporización entre los módulos. Mejor dicho, es el que permite que no haya colisión de información en el sistema.

Existen dos grandes tipos clasificados por el método de envío de la información: **bus paralelo** o **serial**.

Hay diferencias en el desempeño y hasta hace unos años se consideraba que el uso apropiado dependía de la longitud física de la conexión: para cortas distancias el bus paralelo, para largas el serial.

Bus paralelo

Es un bus en el cual los datos son enviados por bytes al mismo tiempo, con la ayuda de varias líneas que tienen funciones fijas. La cantidad de datos enviada es bastante grande con una frecuencia moderada y es igual al ancho de los datos por la frecuencia de funcionamiento. En los computadores ha sido usado de manera intensiva, desde el bus del procesador, los buses de discos duros, tarjetas de expansión y de vídeo, hasta las impresoras.

Bus serie

En este los datos son enviados, bit a bit y se reconstruyen por medio de registros o rutinas de software. Está formado por pocos conductores y su ancho de banda depende de la frecuencia. Es usado desde hace menos de 10 años en buses para discos duros, tarjetas de expansión y para el bus del procesador.