

DIRECCIÓN GENERAL DE EDUCACIÓN SUPERIOR TECNOLÓGICA

INSTITUTO TECNOLÓGICO DE CIUDAD VALLES

DEPARTAMENTO DE SISTEMAS Y COMPUTACIÓN

INGENIERÍA EN SISTEMAS COMPUTACIONALES

LENGUAJES Y AUTOMATAS ll

**GENERACIÓN DE CÓDIGO OBJETO**

**UNIDAD 4**

ELABORADO POR

|  |  |
| --- | --- |
| ORLANDO MEDINA ALONSO 09690035 |  |
| certified sgc rsgc |  |

Contenido

[Introducción 1](#_Toc405403673)

[4 Generación de código objeto. 2](#_Toc405403674)

[4.1.1 Registros. 2](#_Toc405403675)

[4.1.2 Lenguaje Ensamblador. 3](#_Toc405403676)

[4.1.3 Lenguaje Máquina. 7](#_Toc405403677)

[4.1.4 Administración de memoria. 10](#_Toc405403678)

[5 Conclusión. 11](#_Toc405403679)

[6 Bibliografía. 12](#_Toc405403680)

## Introducción

Los lenguajes de programación de bajo nivel son muy apegados a la máquina, algunas empresas de antivirus y videojuegos, siguen programando pequeñas rutinas en ensamblador por su rapidez y a pegamiento al computador. Esto es debido, a que un programa en ensamblador, es ejecutado casi directamente por el computador, ya que hablan casi "el mismo idioma".   
  
Hoy en día es un oficio minucioso de artesanía... sería como el carpintero de lápiz en oreja que va al bosque, tala el árbol, lo lleva al taller, lo pule, lima, da forma y saca el mueble.   
  
En este documento vamos a conocer algunos conceptos básicos del lenguaje ensamblador creo que es esencial para cualquier informático.

# Generación de código objeto.

### Registros.

Una de las consideraciones a tener para el generador de código consiste en la asignación de registros. Los registros son los elementos más valiosos y escasos en la fase de generación de código, puesto que el CPU solamente puede procesar datos que se encuentren en registros.

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

1) Durante la asignación de registros, se selecciona el conjunto de variables y/o constantes que residirán en los registros en un momento del programa.

2) Durante una fase posterior a la anterior, se escoge el registro específico en el que residirá una variable.

### Lenguaje Ensamblador.

El lenguaje Assembly 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 computadoras legible por un programador. Fue usado ampliamente en el pasado para el desarrollo de software, pero actualmente sólo se utiliza en contadas ocasiones, especialmente cuando se requiere la manipulación directa del hardware o se pretenden rendimientos inusuales de los equipos.

Un ensamblador crea código objeto traduciendo instrucciones mnemónicas a códigos operativos, e interpretando los nombres simbólicos para direcciones de memoria y otras entidades. El uso de referencias simbólicas es una característica básica de los ensambladores, evitando tediosos cálculos y direccionamiento manual después de cada modificación del programa. La mayoría de los ensambladores también incluyen facilidades para crear macros, a fin de generar series de instrucciones cortas que se ejecutan en tiempo real, en lugar de utilizar subrutinas. Los ensambladores son por lo general más fáciles de programar que los compiladores de lenguajes de alto nivel, y han estado disponibles desde la década de 1950. Los ensambladores modernos, especialmente para arquitecturas basadas en RISC, como por ejemplo MIPS, SPARC y PA-RISC optimizan las instrucciones para explotar al máximo la eficiencia de segmentación del CPU.

Los ensambladores de alto nivel ofrecen posibilidades de abstracción que incluyen:

Control avanzado de estructuras. Procedimientos de alto nivel, declaración de funciones. Tipos de datos que incluyen estructuras, registros, uniones, clases y conjuntos. Sofisticado procesamiento de macros. Lenguaje [editar]Un programa escrito en lenguaje Assembly consiste en una serie de instrucciones que corresponden al flujo de órdenes ejecutables que pueden ser cargadas en la memoria de una computadora. La transformación del lenguaje Assembly en código máquina la realiza un programa ensamblador, y la traducción inversa la puede efectuar un desensamblador. A diferencia de los lenguajes de alto nivel, aquí hay usualmente una correspondencia 1 a 1 entre las instrucciones simples del ensamblador y el lenguaje máquina. Sin embargo, en algunos casos, un ensamblador puede proveer “pseudo- instrucciones” que se expanden en un código de máquina más extenso a fin de proveer la funcionalidad necesaria. Por ejemplo, para un código máquina condicional como “si X mayor o igual que”, un ensamblador puede utilizar una pseudo-instrucción al grupo “haga si menor que”, y “si = 0″ sobre el resultado de la condición anterior. Los ensambladores más completos también proveen un rico lenguaje de macros que se utiliza para generar código más complejo y secuencias de datos.

Cada arquitectura de computadoras tiene su propio lenguaje de máquina, y en consecuencia su propio lenguaje Assembly. Los ordenadores difieren en el tipo y número de operaciones que soportan; también pueden tener diferente cantidad de registros, y distinta representación de los tipos de datos en memoria. Aunque la mayoría de las computadoras son capaces de cumplir esencialmente las mismas funciones, la forma en que lo hacen difiere, y los respectivos lenguajes Assembly reflejan tal diferencia.

Pueden existir múltiples conjuntos de mnemónicos o sintaxis de Assembly para un mismo conjunto de instrucciones, instanciados típicamente en diferentes programas ensamblador. En estos casos, la alternativa más popular es la provista por los fabricantes, y usada en los manuales del programa.

Las operaciones disponibles en la mayoría de los conjuntos de instrucciones incluye:

mover llenar un registro con un valor constante mover datos de una posición de memoria a un registro o viceversa escribir y leer datos de dispositivos computar sumar, restar, multiplicar o dividir los valores de dos registros, colocando el resultado en uno de ellos o en otro registro realizar operaciones binarias, incluyendo operaciones lógicas (AND/OR/XOR/NOT) comparar valores entre registros (mayor, menor, igual) afectar el flujo del programa saltar a otra posición en el programa y ejecutar instrucciones allí saltar si se cumplen ciertas condiciones (IF) saltar a otra posición, pero guardar el punto de salida para retornar (CALL, llamada a subrutinas) Algunas computadoras incluyen instrucciones complejas dentro de sus capacidades. Una sola instrucción compleja hace lo mismo que en otras computadoras puede requerir una larga serie de instrucciones, por ejemplo:

salvar varios registros en la pila de una sola vez mover grandes bloques de memoria operaciones aritméticas complejas o de punto flotante (seno, coseno, raíz cuadrada ) El nivel de lenguaje Assembly tiene aspectos importantes de los niveles de micro arquitectura, en los cuales se encuentra (ISA y sistema operativo) estos dos se utilizan para la traducción en lugar de la interpretación. Algunas características del lenguaje se describen a continuación Los programas que sirven para traducir algún programa para el usuario se llama traductores, el lenguaje en que está escrito el programa original se llama lenguaje fuente, el lenguaje original que sea modificado se llama lenguaje objeto.

Se usa la traducción cuando se cuenta con un procesador (ya sea hardware o un intérprete) para el lenguaje objeto pero no para el lenguaje fuente, Si la traducción se realiza correctamente, la ejecución del programa traducido dará exactamente los mismos resultados que habría dado la ejecución del programa fuente. Hay dos diferencias entre traducción e interpretación, en la traducción no se ejecuta directamente el programa original, en el lenguaje fuente se convierte en un programa equivalente llamado programa objeto o programa binario ejecutable y este funciona solo cuando se ha acabado la traducción.

#### Ejemplo

Un procesador x86 puede ejecutar la siguiente instrucción binaria como se expresa en código maquina:

**Binario:** 10110000 01100001 (Hexadecimal: 0xb061) La representación equivalente en Assembly es más fácil de recordar: mov al, 061h

Esta instrucción significa:

Mueva el valor hexadecimal 61 (97 decimal) al registro “al”. El mnemónico “mov” es un código de operación u “opcode”, elegido por los diseñadores de la colección de instrucciones para abreviar “mov” (mover).- El opcode es seguido por una lista de argumentos o parámetros, completando una instrucción de ensamblador típica.

### Lenguaje Máquina.

Un lenguaje de programación de bajo nivel es el que proporciona poca o ninguna abstracción del microprocesador de un ordenador. Consecuentemente es fácilmente trasladado a lenguaje de máquina. La palabra “bajo” no implica que el lenguaje sea inferior a un lenguaje de alto nivel; se refiere a la reducida abstracción entre el lenguaje y el hardware.

**Uso:** ventajas e inconvenientes. En general se utiliza este tipo de lenguaje para programar controladores (drivers). La programación en un lenguaje de bajo nivel como el lenguaje de la máquina o el lenguaje simbólico tiene ciertas ventajas:

* Mayor adaptación al equipo.
* Posibilidad de obtener la máxima velocidad con mínimo uso de memoria.
* Pero también tiene importantes inconvenientes:
* 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 intrucciones.
* Es necesario conocer en detalle la arquitectura de la máquina.

#### Primera generación

El lenguaje de programación de primera generación (por sus siglas en inglés, 1GL), es el lenguaje de código máquina. Es el único lenguaje que un microprocesador entiende de forma nativa. El lenguaje máquina no puede ser escrito o leído usando un editor de texto, y por lo tanto es raro que una persona lo use directamente.

#### Segunda generación

El lenguaje de programación de segunda generación (por sus siglas en inglés, 2GL), es el lenguaje ensamblador. Se considera de segunda generación porque, aunque no es lenguaje nativo del microprocesador, un programador de lenguaje ensamblador debe conocer la arquitectura del microprocesador (como por ejemplo las particularidades de sus registros o su conjunto de instrucciones).

#### Direccionamiento Lenguaje Maquina

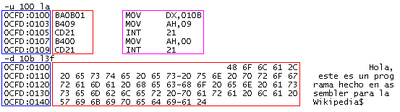
Es la forma en cómo se accede a la memoria. Recordar que un programa no puede ejecutarse sino se encuentra en memoria principal. La forma de acceder a la memoria depende del microprocesador, pero en general existen dos tipos de direccionamiento: directo e indirecto.

El direccionamiento directo también recibe el nombre de direccionamiento absoluto y el acceso a las direcciones se hace de manera directa. El 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 localizables e independientes. Si la dirección base es el inicio de la memoria fija el direccionamiento pasa a ser un variante de direccionamiento absoluto.

#### Ejemplos.

Lenguaje de máquina del [Intel 8088](http://es.wikipedia.org/wiki/Intel_8086_y_8088). El código de máquina en [hexadecimal](http://es.wikipedia.org/wiki/Sistema_hexadecimal) se resalta en rojo, el equivalente en [lenguaje ensamblador](http://es.wikipedia.org/wiki/Lenguaje_ensamblador) en magenta, y las direcciones de memoria donde se encuentra el código, en azul. Abajo se ve un texto en hexadecimal y [ASCII](http://es.wikipedia.org/wiki/ASCII).



Todas las instrucciones preparadas en cualquier lenguaje de máquina tienen por lo menos dos partes. La primera es el comando u operación, que dice a la computadora cuál es la función que va a realizar.



### 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 los punteros y tienen un programa denominado recolector de basura que se encarga de limpiar la memoria no utilizada mejorando el desempeño

# Conclusión.

Lenguaje de la máquina que se utiliza en binario el más usamos son los lenguajes naturales que son más sencillos para su comprensión este lenguaje de maquina tiene como características que hace operaciones aritméticas lógicas y de control, también depende de la arquitectura de la máquina para su realización. Direccionamiento este donde se va ejecutar el programa en cual debe estar almacenado en la memoria principal puede ser directa o indirecta es cuando se inicia la ejecución del programa. Lenguaje ensamblador es un traductor de códigos de baja nivel está directamente en la máquina para su uso. Característica es que se puede utilizar en su máquina. Ensambladores básicos solo ejecuta símbolos y instrucciones del direccionamiento del mismo. Ensambladores moduladores son lenguajes de alto nivel. Almacenamiento no permite que maneje la memoria de la máquina. Registros son las instrucciones que guardan en la memoria distribución es que se puede ejecutar o usar en oras maquinas.

# Bibliografía.

Consulta a la página [Web](http://www.monografias.com/trabajos5/laweb/laweb.shtml): http://programaciondesistemasito.blogspot.mx/2013/10/unidad-8-generacion-de-codigo-objeto.html

Autor: [YESENIA CETINA](http://es.slideshare.net/yesyduc10?utm_campaign=profiletracking&utm_medium=sssite&utm_source=ssslideview), DOCENTE ITSVA Fecha de actualización: domingo 13 de octubre 2013

Consulta a la página [Web](http://www.monografias.com/trabajos5/laweb/laweb.shtml): http://www.itescam.edu.mx/principal/sylabus/fpdb/recursos/r99909.PDF

Autor: LIC. YVES GERAUD MENDOZA Fecha de actualización: 3/07/2010 - 12:31

Consulta a la página [Web](http://www.monografias.com/trabajos5/laweb/laweb.shtml): http://silvamondragon.wordpress.com/2010/11/29/4-2-4-lenguaje-ensamblador/

Autor: Rafael Vázquez Pérez Fecha de actualización: 23/09/2012 - 16:52