

Materia: Lenguajes autómatas 2

Gabriel Iván García Palacios 13480162



Viernes 4 de Mayo 2018



GENERACIÓN DE CÓDIGO OBJETO

La fase final de un compilador es la generación de código objeto, que por lo general consiste en código de máquina relocizable o código ensamblador. Las posiciones de memoria se seleccionan para cada una de las variables usadas por el programa. Después, cada una de las instrucciones intermedias se traduce a una secuencia de instrucciones de máquina que ejecuta la misma tarea. Un aspecto decisivo es la asignación de variables a registros.

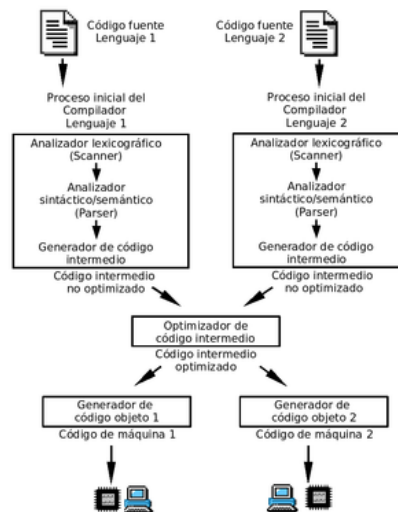


fig 1. Estructura de un compilador

REGISTROS

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.

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.

Que hacen

Los registros del procesador se emplean para controlar instrucciones en ejecución, manejar direccionamiento de memoria y proporcionar capacidad aritmética. Los registros son direccionables por medio de una viñeta, que es una dirección de memoria. Los bits, por conveniencia, se numeran de derecha a izquierda (15, 14, 13, ..., 3, 2, 1, 0).

Tipos

- . Registros de segmento
- . Registros de apuntadores de instrucciones
- . Registros apuntadores
- . Registros de propósitos generales
- . Registro índice
- . Registro de bandera

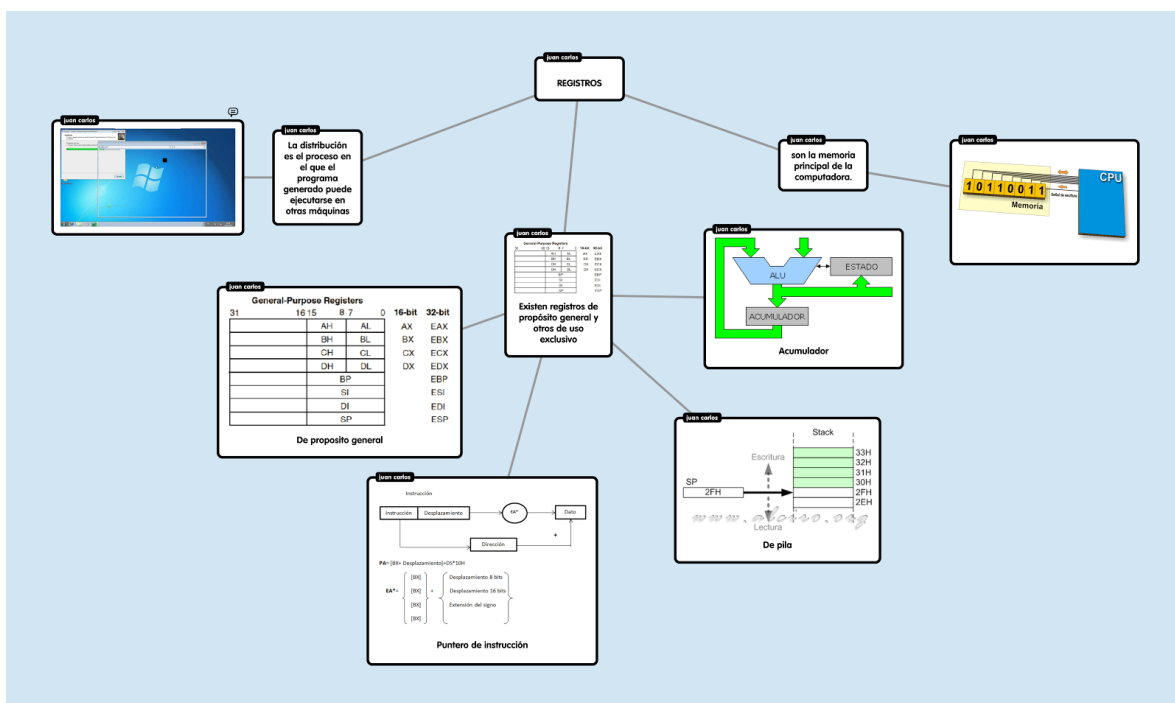


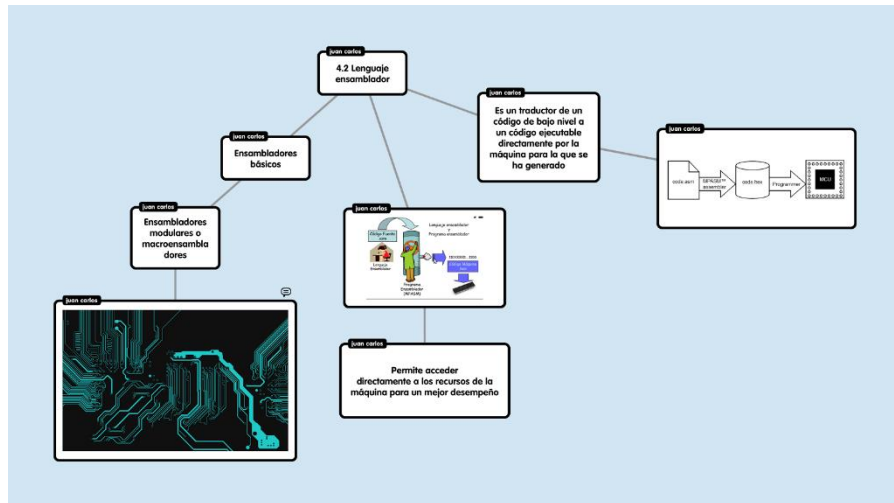
Fig2



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. Que 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. En 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 incluye facilidades para crear macros, así como 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 1912. Los ensambladores modernos, especialmente para arquitecturas basadas en 32 bits, como por ejemplo 64 bits, 32 bits y 16 bits optimizan las instrucciones para explotar al máximo la eficiencia de segmentación del CPU.

Cada arquitectura de computadoras tiene su propio lenguaje de máquina, y en consecuencia su propio lenguaje Assembly. Los ordenadores difieren en el tipo, 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.



Lenguaje maquina

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 instrucciones.

?



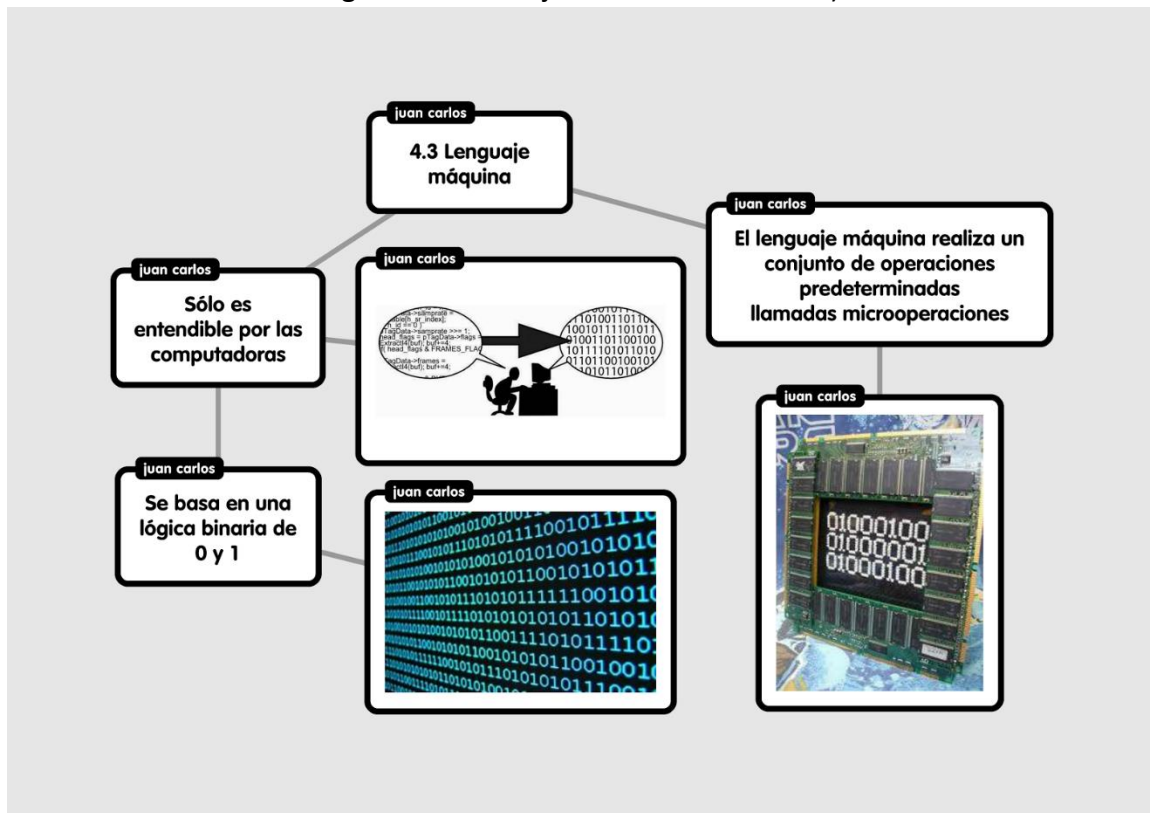
Es necesario conocer en detalle la arquitectura de la máquina

4.1.3.1 Primera generación

El lenguaje de programación de primera generación por sus siglas en inglés, (IL), 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.

4.1.3.2 Segunda generación

El lenguaje de programación de segunda generación por sus siglas en inglés, (■IL), 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).



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 mayor! A 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 máquina 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.

Bibliografia

Consulta a la página Web:

<http://programaciondesistemasito.blogspot.mx/2013/10/unidad-8- generacion-de-codigo-obeto.html>!utor: "

