

# **INSTITUTO TECNOLÓGICO DE NUEVO LEÓN**



## **“PROYECTO 4”**

**ALUMNO:** Iván Alejandro Padilla Esparza  
#15480063

**CATEDRÁTICO:** Juan Pablo Rosas

Guadalupe, Nuevo León a 03 de mayo del 2018.

## Introducción.

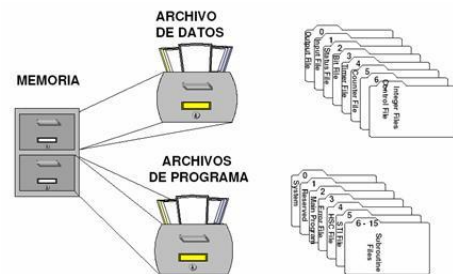
En este reporte vamos a conocer algunos conceptos básicos del lenguaje ensamblador y máquina. Conoceremos del código objeto, este es la última fase de un compilador y consiste en transformar un código a un código de bajo nivel.

Así mismo la importancia de los registros y sus tipos. El almacenamiento de memoria, etc.

Vamos a conocer un poco más a fondo los lenguajes ensamblador y máquina, la ayuda que estos generan sobre el código y la manera de facilitar a la máquina para ser más comprensible su lectura.

## Capítulo 1: Registro.

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.



### **Distribución**

La distribución es el proceso en el que el programa generado puede ejecutarse en otras máquinas. Con respecto al ensamblador, la mayoría del direccionamiento se hace relativo para que el programa sea relocable por un programa llamado cargador. En el caso de programas compilados se necesitan de las librerías, si son estáticas se incluyen en el ejecutable por lo que el programa se hace gráfico, si son dinámicas no pero el programa es más pequeño.

Los registros se pueden clasificar de la siguiente forma:

**Registros de datos**

**Registros de segmentos**

**Registros punteros de pila**

**Registros índices**

### ***¿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*

Debido a la complejidad del software actual se necesitan de asistentes para poder instalar y ejecutar un programa. Operar sobre registros es más rápido y eficiente que operar sobre memoria. Por ello, la adjudicación eficiente de registros tiene un gran impacto en la performance.

El uso de registros puede dividirse en dos subproblemas:

- Durante la reserva de registros (allocation), se seleccionan el conjunto de variables que vivirá en registros en un punto del programa.
- Durante la (posterior) asignación de registros (assignment), se elige el registro específico para cada variable.

## Capítulo 2: Lenguaje ensamblador.

El ensamblador, es un traductor de un código de bajo nivel a un código, ejecutable directamente por la máquina para la que se ha generado. Fue la primera abstracción de un lenguaje de programación, posteriormente aparecieron los compiladores.

### *¿Dónde se utiliza el lenguaje ensamblador?*

El lenguaje ensamblador trabaja directamente con el microprocesador de una máquina, es decir, programamos un microprocesador con este lenguaje, también lo podemos utilizar para desarrollar controladores de dispositivos (Drivers) y sistemas operativos de computadoras.

### *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.

### *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, antes de la generalización de los lenguajes de alto nivel. Un macroinstrucción es el equivalente a una función en un lenguaje de alto nivel.

### *Ventajas.*

Una de las principales ventajas del uso del ensamblador, es que se encarga de administrar de manera transparente para el usuario la creación de memoria, las bifurcaciones y el paso de parámetros.

### **Ejemplo 1**

El siguiente es un ejemplo del programa clásico Hola mundo escrito para la arquitectura de procesador x86 (bajo el sistema operativo DOS) en modo texto (por defecto).

```
programa:
    mov ax, @data
    mov ds, ax
    mov dx, offset Cadena1
    mov ah, 9
    int 21h
    int 20h
end programa

.model small
.stack
.data
Cadena1 DB 'Hola Mundo.$'
.code
```

## Capítulo 3: Lenguaje máquina.

El **lenguaje máquina** es el único que entiende la computadora digital, es su "lenguaje natural". En él sólo se pueden utilizar dos símbolos: 0 y 1. Por ello, al lenguaje máquina también se le denomina *lenguaje binario*. La computadora sólo puede trabajar con bits, sin embargo, para el programador no resulta fácil escribir instrucciones tales como:

10100010

11110011

00100010

00010010

### **Características**

- El lenguaje máquina realiza un conjunto de operaciones predeterminadas llamadas microoperaciones.
- Las microoperaciones sólo realizan operaciones del tipo aritmética (+, -, \*, /), lógicas (AND, OR, NOT) y de control (secuencial, decisión, repetitiva).
- El lenguaje máquina es dependiente del tipo de arquitectura. Así un programa máquina para una arquitectura Intel x86 no se ejecutará 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 del Lenguaje Máquina.**

Posibilidad de cargar (transferir un programa a la memoria) sin necesidad de traducción posterior, lo que supone una velocidad de ejecución superior a cualquier otro lenguaje de programación.

### **Desventajas del Lenguaje Máquina.**

Dificultad y lentitud en la codificación.  
Poca fiabilidad.  
Gran dificultad para verificar y poner a punto los programas.  
Los programas solo son ejecutables en el mismo procesador (CPU).

### **Direccionamiento**

El direccionamiento directo, el acceso a las direcciones se hace de manera directa, se basa a partir de una dirección genérica, generalmente el inicio del programa.

El direccionamiento relativo hace a los programas re localizables e independientes. Si la dirección base es el inicio de la memoria fija el direccionamiento pasa a ser una variante de direccionamiento absoluto.

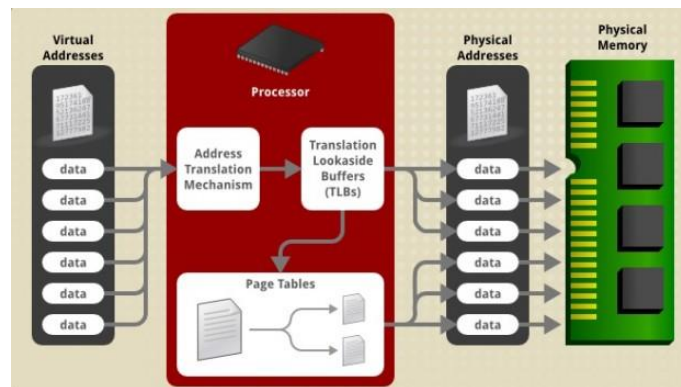
## Capítulo 4: Administración de memoria.

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.

Consiste en determinar la posición de memoria en la que los diferentes símbolos del programa almacenan la información

Depende de la estrategia utilizada para la gestión de memoria, el mecanismo puede variar.

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.



### **Compartimiento.**

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.

### **Organización física.**

Debido al costo de una memoria principal rápida, ésta 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.

## Conclusión.

Las instrucciones que guardan en la memoria distribución es que se puede ejecutar o usarlas en diferentes computadores, son los registros. Estos son de suma importancia al momento de generar código, ya que el CPU solamente puede procesar datos que se encuentren en dichos registros.

La fase final de un compilador es conocida como código objeto, este mismo consiste en transformar el código intermedio en código de bajo nivel, con la finalidad de ser más entendible para la máquina, puede ser en lenguaje máquina o ensamblador.

El lenguaje máquina y ensamblador gracias a su traducción de código, instrucciones, ayudan a las máquinas a comprender y así realizar programas más complejos/alto nivel. Nos ayuda a pasar a un código a código de bajo nivel, para que así sea ejecutado correctamente.

Las computadoras solo pueden entender el lenguaje máquina, ya que se basa en una lógica binaria de 0 y 1, generalmente implementada por mecanismos eléctricos.

## Conceptos.

**Registros:** Es un tipo de dato estructurado con un número fijo de componentes (no todos necesariamente del mismo tipo) a los que se accede por su nombre.

**Registros de datos:** Se usan para guardar números enteros. En algunas computadoras antiguas, existía un único registro donde se guardaba toda la información, llamado *acumulador*.

**Registros de segmentos:** 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:** Se utiliza para fijar el puntero de pila y así poder acceder a los elementos de la pila.

**Registros índices:** Índice fuente e Índice destino.

**Lenguaje ensamblador:** Consiste en un conjunto de mnemónicos que representan instrucciones básicas para los computadores, microprocesadores, microcontroladores y otros circuitos integrados programables.

**Lenguaje maquina:** Es el sistema de códigos directamente interpretable por un circuito micro programable, como el microprocesador de una computadora o el microcontrolador de un autómatas.

**Memoria:** Es el dispositivo que retiene, memoriza o almacena datos informáticos durante algún período de tiempo.

**Multiprogramación:** Una técnica por la que dos o más procesos pueden alojarse en la memoria principal y ser ejecutados concurrentemente por el procesador o CPU.

**Compartimiento:** En lugar de asignar una cantidad dedicada de memoria física a cada partición que utiliza memoria compartida.

**Microoperaciones:** Es una operación básica realizada sobre la información almacenada en uno o más registros (flip-flops). El resultado de la operación puede sustituir la información binaria anterior de un registro o puede transferirse a otro.



## Bibliografía.

Registros, lenguaje ensamblador, maquina, escrito en sitio Web:

<http://acaurio.blogspot.mx/>

Lenguaje máquina y ensamblador, escrito en sitio web:

<http://arquitecturadelcomputadorsemestre6.blogspot.mx/2014/10/lenguaje-maquina-y-lenguaje-ensamblador.html>

Lenguaje ensamblador, artículo en:

<http://silvamondragon.wordpress.com/2010/11/29/4-2-4-lenguajeensamblador/>

## Reporte.

En general el lenguaje máquina es difícil de entender para los humanos por este motivo hacemos uso de lenguajes más parecidos a los lenguajes naturales (son los que más usamos, ya que son más sencillos de comprender).

El 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.

Nos hacen mención que la administración de la memoria es un proceso que hoy en día es muy importante, ya que hacer mal o buen uso tiene una acción directa sobre el desempeño de memoria.

También nos habla del direccionamiento, aquí es donde se ejecuta el programa, dicho programa debe estar almacenado en la memoria principal, puede ser directa o indirecta.

Del lenguaje ensamblador podemos decir que es un traductor de códigos de baja nivel, este lenguaje esta específicamente en la máquina para su uso. Dicho lenguaje consta de característica es que se puede utilizar en su máquina, en el resumen pudimos mostrar 2, la cual son “ensambladores básicos” (estos solos ejecutan símbolos e instrucciones) y “ensambladores modulares” (estos son lenguajes de alto nivel).