



Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ciencias y Sistemas

Curso: Arquitectura y Ensambladores 1
Plataforma: Windows



Uso del Leguaje Ensamblador

Descripción:

Este tutorial es una guía para lograr entender los principales comandos de línea de DOS las cuales el lenguaje ensamblador utiliza para controlar la computadora. Además de esto daremos una guía para la instalación de Borland Turbo Assembler 5.0, y la forma de utilizarlo.



Índice

Introducción	2
Objetivos	3
Lenguaje Ensamblador	4
Historia	4
Definición	4
Instalacion Borland Turbo Assembler 5.0	6
Compilación de un ejemplo en Borland Turbo Assembler 5.0	10
Funcionamiento	12
Los Registros	12
Direccionamiento en modo real	15
Instrucciones para el movimiento de datos	16
Instrucciones de transferencia de cadenas de datos	17
Instrucciones para operaciones aritméticas	18
Instrucciones para Interrupciones	20
Ejemplo Práctico	24
Conclusiones	26
Bibliografía	27
Recomendaciones	28



Introducción

El lenguaje ensamblador es un tipo de lenguaje de bajo nivel, se llaman así porque están muy cercanos al hardware del ordenador. Es por ello de el lenguaje ensamblador asigna una abreviatura a cada instrucción en binario (código máquina), de esta forma es más fácil recordar y más difícil equivocarse con las instrucciones o abreviaturas.

Una desventaja es que con este lenguaje sigue siendo necesario conocer muy bien el hardware del ordenador. En otras palabras es necesario conocer a fondo la arquitectura de la maquina para la que se va a programar.

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.



Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ciencias y Sistemas

Objetivos

General

- Comprender el Lenguaje Ensamblador.

Específicos

- Realizar la instalación adecuada de de Borland Turbo Assembler 5.0.
- Desarrollo eficiente de programas en lenguaje ensamblador.



Leguaje Ensamblador

1. Historia [1]

Los lenguajes de computadoras toman diferentes formas; los de las primeras computadoras, como la ENIAC y la EDSAC, se componían en el lenguaje real de las máquinas mismas. La dificultad de programar las máquinas de esta manera limitaba drásticamente su utilidad y proporcionaba un fuerte incentivo para que se desarrollaran lenguajes de programación más orientados hacia la expresión de soluciones con la notación de los problemas mismos.

Los primeros lenguajes de programación se conocieron como Lenguajes Ensambladores, un ejemplo es: TRANSCODE, desarrollado para la computadora FERUT. En los lenguajes ensambladores se define un código especial llamado mnemónico para cada una de las operaciones de la máquina y se introduce una notación especial para especificar el dato con el cual debe realizarse la operación.

2. Definición [2]

Un programa escrito en lenguaje ensamblador consiste en una serie de instrucciones que corresponden al flujo de órdenes ejecutables que pueden ser cargadas en la memoria de una computadora. Por ejemplo, un procesador x86 puede ejecutar la siguiente instrucción binaria como se expresa en código máquina:

Binario: 10110000 01100001 (Hexadecimal: 0xb061)

*La representación equivalente en ensamblador 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 "move" (mover).- El opcode es seguido por una lista de argumentos o parámetros, completando una instrucción de ensamblador típica.

La transformación del lenguaje ensamblador 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 pseudoinstrucción 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 ensamblador. 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 ensambladores reflejan tal diferencia.



Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ciencias y Sistemas

Pueden existir múltiples conjuntos de mnemónicos o sintaxis de ensamblador 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.

❖ **Ventajas**

Máximo rendimiento: máxima velocidad y mínimo tamaño
Puede explotar al máximo el CPU
Acceso a cualquier recurso de la computadora
Ayuda a conocer a detalle cómo trabaja la computadora

❖ **Desventajas**

El programa sólo corre en un CPU
Es difícil llegar a ser programador experto
Es difícil escribir programas grandes y complejos
Es difícil depurar y dar mantenimiento a los programas
Un error puede provocar que falle el equipo



Instalacion Borland Turbo Assembler 5.0

Ahora procederemos a descargar Borland Turbo Assembler 5.0 del sitio web <http://vetusware.com/download/TASM%205.0?id=65>

A continuación debe dar clic sobre el boton *download*. Aparecera la ventana de descarga, donde usted le indicara la direccion donde lo guardara, ej *C:\Download\Assembler*. Este proceso puede tardar varios minutos. El instalador viene comprimido en formato *.rar* si no cuenta con un programa para descomprimirlo, puede descargarlo del sitio web http://www.programas-gratis.net/php/programa2.php?id_programa=570.

Ahora procederemos a descomprimir el instalador de la siguiente forma:

Damos clic derecho sobre el *Tasm_5.rar*, y luego seleccionamos la opcion "Extraer en Tasm_5"



Figura1. Extraer archivos

Aparecerá una carpeta con el nombre *Tasm_5* con los archivos descomprimidos. Luego de esto abrimos esta carpeta y debemos dar clic sobre "*Instal*", el cual iniciara la instalación:

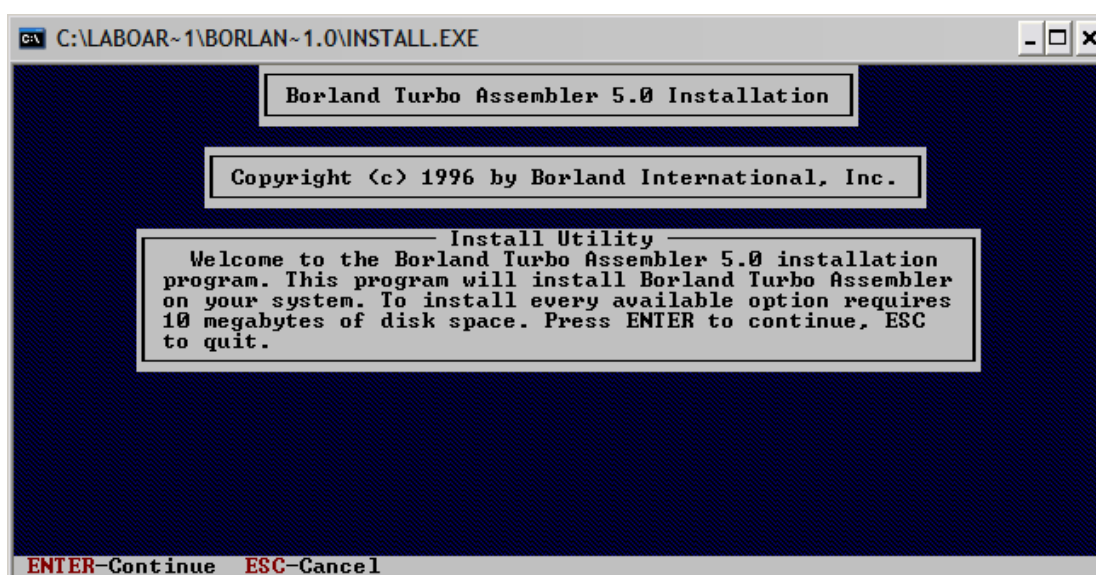


Figura2. Pantalla inicial de Instalacion



Luego presionamos *Enter* para continuar. Aparecera la siguiente ventana:

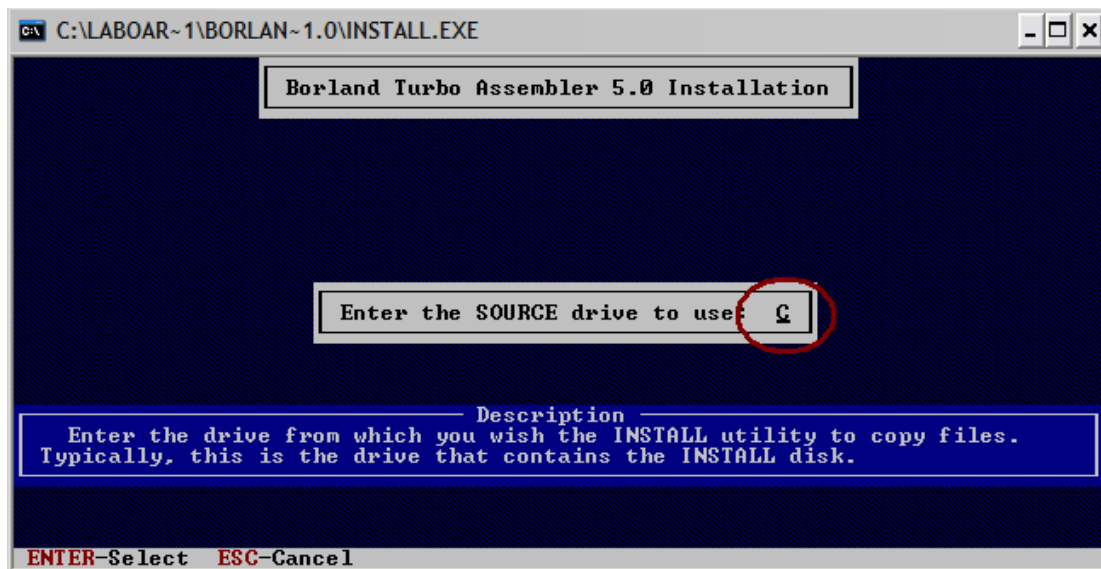


Figura3. Pantala de Configuración

En esta ventana debemos cambiar el lugar donde se instalaran nuestros archivos, en este caso escogimos el disco C. Luego de esto presionamos "*Enter*" para continuar con la instalacion.

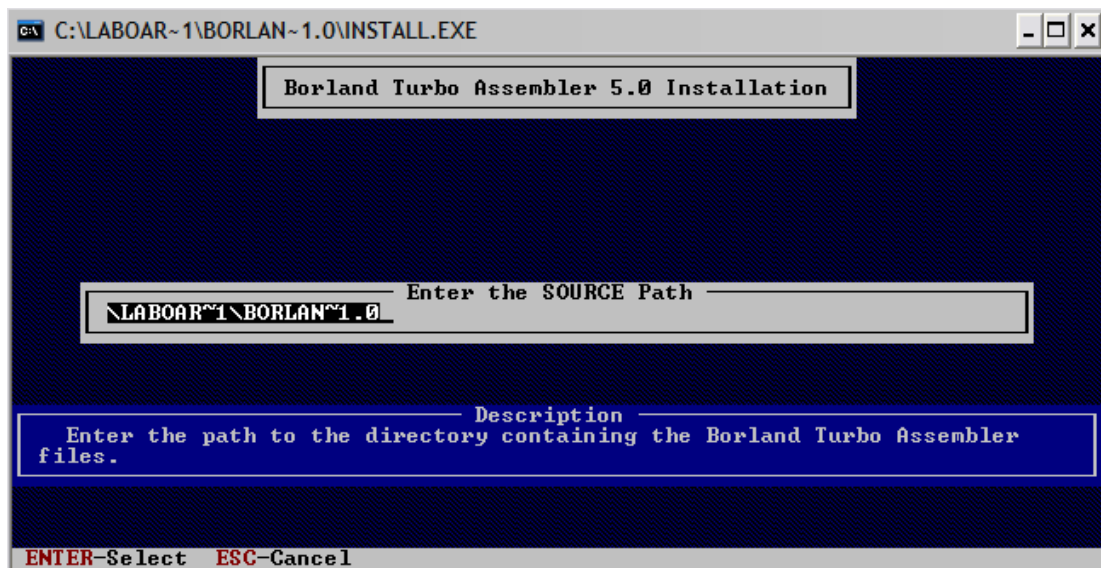


Figura4. Pantalla de Instalacion

En esta ventana nos pide la direccion de los archivos que descomprimimos previamente, solamente damos "*Enter*" dejando la que aparece por default.



Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ciencias y Sistemas

Luego aparecera la siguiente ventana, en ella deje todas las opciones tal y como estan por default, y seleccione la opcion "stara Installation"y luego de "Enter".

Otra forma de iniciar la instalacion es presionando F9.

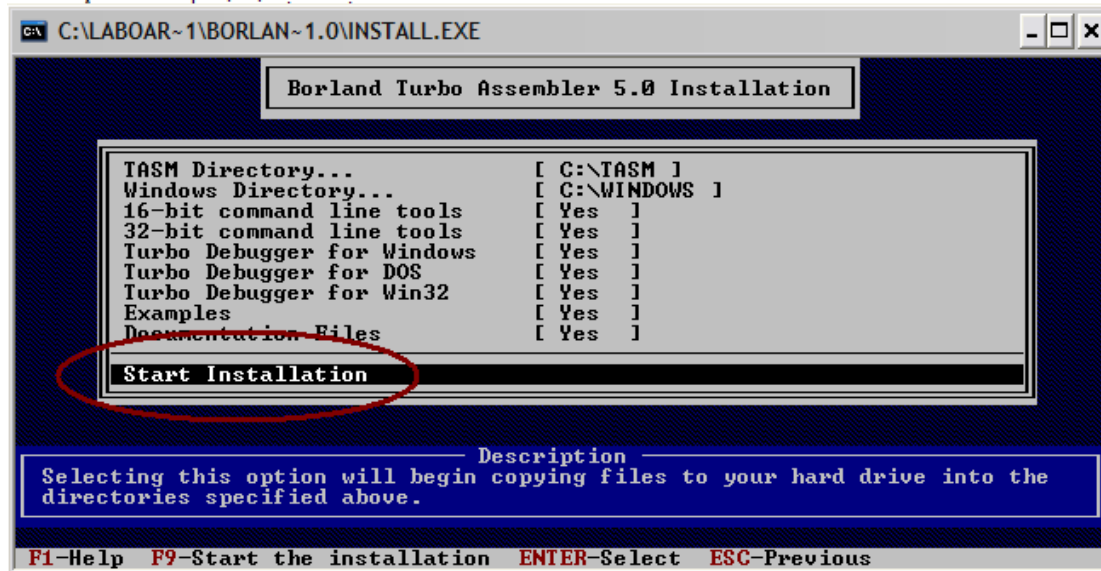


Figura5. Pantalla de Configuración de la Instalacion

Luego aparecera un recuadro dando el estado de la instalacion. Debe esperar unos minutos para terminar la instalacion.

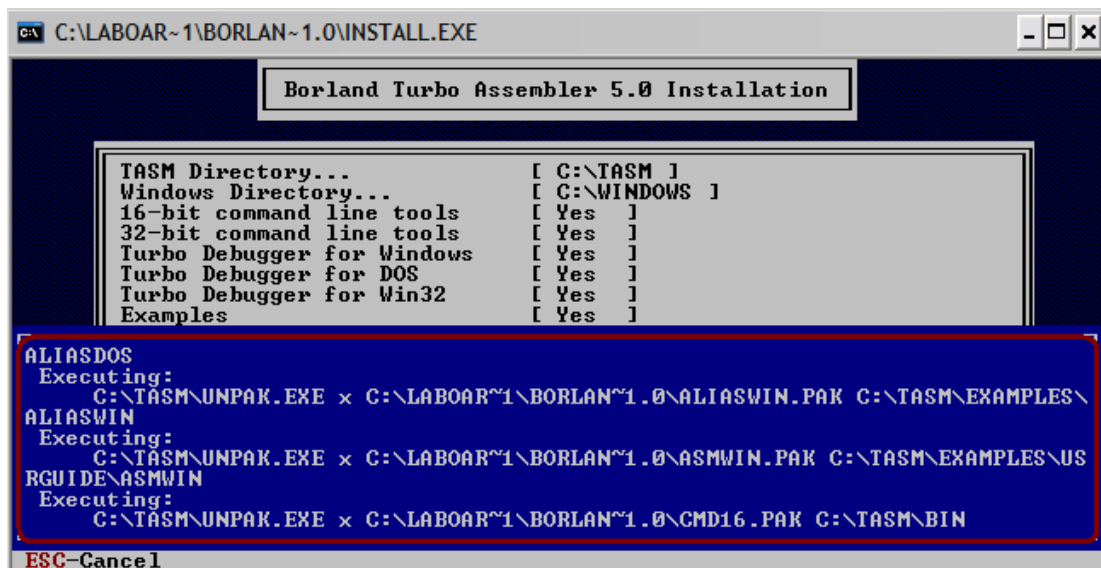


Figura6. Pantalla Proceso de Instalacion



Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ciencias y Sistemas

Por ultimo, y no por eso menos importante aparecerá el ReadMe de Assembler, le recomendamos ver el contenido ya que contiene ayuda sobre el programa.

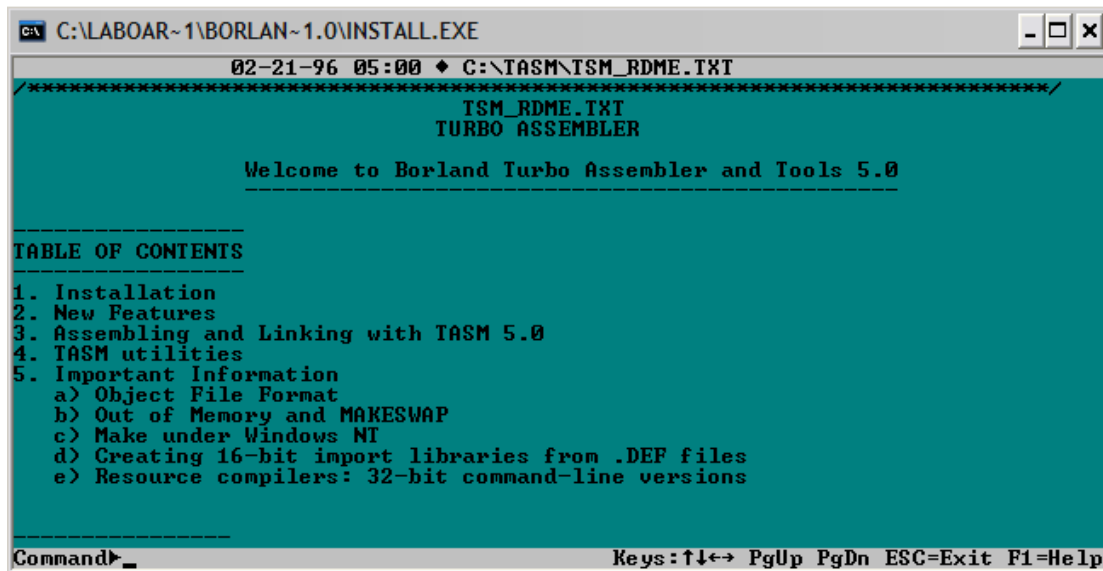


Figura7. Leeme (Readme) de Assembler

Para salir de esta pantalla presione ESC.

Compilación de un ejemplo en Borland Turbo Assembler 5.0

El código que compilaremos es un programa que utiliza a CX utilizado como Contador de 9 a 0, el código es el siguiente:

```
.model small
.stack
.data
    TEXTO db 10,'CX utilizado como Contador',13,10,'$'
    contador dw 58,13,10,'$'
.code
inicio:
    mov ax, @data
    mov ds,ax
    mov dx, offset TEXTO ; Encabezado
    mov ah, 09h
    int 21h
    mov cx,10 ; inicializo
for:
    mov dx, offset contador ; Desplegar en pantalla
    mov ah, 09h ; el valor
    int 21h
    mov ah, 08h ; Entrada del teclado
    int 21h ; Interrupcion
    dec cx ; Disminuyo CX
    dec [contador] ; resto al ascii
    or cx,cx ; seguir mientras
    jnz for ; no sea cero
    mov ah, 09h ; escribo el cero
    int 21h
```



Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ciencias y Sistemas

```
salida:
    mov ax, 4c00h ;Salida del ejecutable
    int 21h
end inicio
```

Ese código lo guardaremos en el siguiente directorio: C:\TASM\BIN con la extensión ASM

Luego abrimos una consola de DOS de Windows: Inicio > Ejecutar > cmd

Y escribimos lo siguiente:

```
Cd c:\tasm\bin
```

```
Tasm Ejemplo
```

```
Tlink Ejemplo
```

Así como se muestra en la siguiente pantalla:

```
C:\WINDOWS\system32\cmd.exe - tlink Ejemplo
Microsoft Windows XP [Versión 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Axelandra Marmol>cd c:\tasm\bin

C:\TASM\BIN>tasm Ejemplo
Turbo Assembler Version 4.1 Copyright (c) 1988, 1996 Borland International

Assembling file:    Ejemplo.ASM
Error messages:     None
Warning messages:   None
Passes:             1
Remaining memory:   412k

C:\TASM\BIN>tlink Ejemplo
Turbo Link Version 7.1.30.1. Copyright (c) 1987, 1996 Borland International

C:\TASM\BIN>
```

Figura8. Ejemplo en Consola

Para ejecutar lo que hemos compilado debe escribir:

```
Ejemplo.exe
```

Y listo, aparecerá la aplicación que halla compilado, siempre y cuando no tenga ningún mensaje de Error.





```
dibuja:                                ;en CX el alto, DX el ancho, SI DI
correspondientes
    CLD
    push    cx                        ; guardo el ancho
    rep     movsb                     ; aqui se vuelve cero CX
    pop     cx
    mov     ax,320                    ; longitud de linea
    sub     ax,cx
    add     di,ax
    dec     dx
    jnz     dibuja
    RET

exit:    mov     ax,0003h             ;text
         int     10h
         mov     ax,4c00h             ;quit
         int     21h

END START
```



Conclusiones

- Para el Lenguaje Ensamblador su funcionamiento radica en la importancia de su uso y su profundo estudio, ya que ayuda a conocer a detalle cómo trabaja un CPU, definitivamente las aplicaciones que se le dan es tanto en la Electrónica y Sistemas.
- Como sabemos el lenguaje ensamblador es el que va traduciendo instrucciones de código de máquina a mnemónicos, es decir interpreta los nombres de los mnemónicos a direcciones de memoria y otras entidades.
- Las instrucciones MOVS, LODS, STOS son bastante útiles cuando se maneja una gran cantidad de datos gracias a que éstas manipulan los índices automáticamente, así como también por la versatilidad del prefijo REP.
- El incremento o disminución de los punteros DI y SI en la transferencia de cadenas dependerá de la posición en memoria donde se encuentre la información, según el programador lo haya especificado.
- El modo segmentado de memoria permite a los programas poder moverse reubicarse en la misma, debido a que no se manejan posiciones completamente directas en memoria sino referentes al valor de segmento.



Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ciencias y Sistemas

Tutorial desarrollado por:

Ronmell Fuentes
Joel Morales Hernandez
Nery Chucuy
Alejandra Molina Martínez

Bibliografía

[1]: http://html.rincondelvago.com/lenguajes-algoritmicos_historia-y-sus-aplicaciones.html

[2]: http://es.wikipedia.org/wiki/Lenguaje_ensamblador

Los microprocesadores Intel
Autor: Barry B. Bray



Recomendaciones

- La mejor herramienta para bajar e instalar a nuestro criterio es Borland Turbo Assembler and Tools 5.0, ya que trae varios complementos o herramientas auxiliares al tema, y aunque estas no fueron tratadas en este tutorial, son una ayuda extra.
- Se le recomienda trabajar con instrucciones de traslado de cadenas cuando utiliza el modo grafico, ya que éstas poseen sencillez y versatilidad al trabajar con grandes cantidades de datos (La RAM de video por ejemplo).
- Hacer prácticas de segmentación de memoria para adquirir soltura en el manejo de los apuntadores y segmentos de informacion cuando programe.