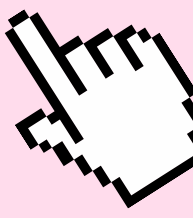
MANUAL BÁSICO DE ENSAMBLADOR

El ABC del principiante



Máquina Virtual

En el curso de arquitectura de computadores se utilizará la máquina virtual DOS BOX para ejecutar programas en ensamblador.

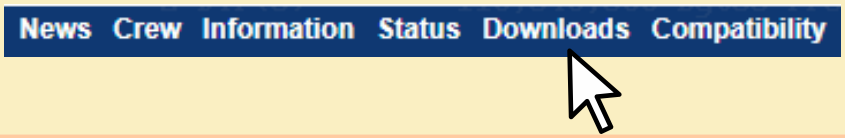
- Favor seguir estos pasos para su instalación:

1

Primero se debe ingresar al siguiente link:

https://www.dosbox.com/

Una vez en la página entraremos a la pestaña de Descargas



2

Dentro de esta pestaña de Descargas, daremos click a este botón:

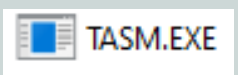
Este botón iniciará la descarga del programa para nuestro correspondiente sistema operativo.

3

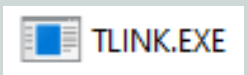
Una vez finalizada la descarga del programa, al abrirlo nos pedirá confirmar algunas especificaciones como confirmar la ubicación del programa para así poder completar la instalación

4

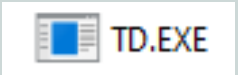
Para poder ejecutar y los programas y hacer uso de las funciones del DOS.BOX, es necesario contar con tres archivos principales :



Para ensamblar los programas



Para enlazar los programas



Para depurar los programas

Segmentos y Datos

En esta sección se explicarán algunos de los registros de segmento necesarios para la creación de un programa en ensamblador. Asimismo se explicará algunos puntos acerca de la declaración de datos.

SEGMENTO DE DATOS

Nmónico: DS
Tamaño: 16 bits

Este es un registro de segmento que posee la información relacionada con la declaración de variables, constantes y define áreas de trabajo o direcciones para poder acceder a cada una de ellas

SEGMENTO DE PILA

Nmónico: SS
Tamaño: 16 bits

Dentro de este resgistro de segmento permite el almacenamiento de datos y direcciones cuyo uso sea de manera temporal. Uno de sus usos puede ser para el caso de aplicación de subrutinas dentro del código

SEGMENTO DE CÓDIGO

Nmónico: CS
Tamaño: 16 bits

En esta sección se escribirán las instrucciones necesarias para que nuestro programa pueda funcionar.

SEGMENTO EXTRA

Nmónico: ES
Tamaño: 16 bits

Tiene funciones específicas como el manejo de cadenas o expandir alguno de los otros segmentos del código, como por el ejemplo el DS.

DECLARACIÓN DE VARIABLES

Para la declaración de variables es necesario la utilización de tres partes:

- Nombre: El nombre de la variable a declarar debe ser único dentro del programa
- Directiva: Esta indica cantidad de bytes a reservar.
  db = define byte
  dw = define word
  dd = define double word
  dq = define quad word
- Inicialización : Valor con el que inicia la variable declarada.

PARTES DE UN PROGRAMA

A CONTINUACIÓN SE ANALIZARÁN ALGUNAS PARTES QUE TODO PROGRAMADOR QUE INICIE EN ENSAMBLADOR DEBE DE CONOCER

REGISTROS

Los registros son espacios reservados de memoria. Estos permiten al programador un mejor manejo de las instrucciones en ejecución al disponer de espacios para almacenar datos y/o direcciones.

Registros de Índice

Registros utilizados para indicar el offset dentro de un segmento.

Colombia

Registros de Índice. SI

Registro que contiene el índice de origen dentro de operaciones que manejen cadenas, arreglos o caracteres. Se asocia con el ES.

Registros de Destino. DI

Registro que contiene el índice de destino dentro de operaciones que manejen cadenas, arreglos o caracteres. Se asocia con el DS.

Registros de Propósito General

Son registros de datos de 32 bits. Poseen dos mitades, inferior y superior, de 8 bits cada una.

Cada uno de estos con un uso específico.

Registro Acumulador. AX

Funciona como el acumulador principal en la mayoría de instrucciones aritméticas del lenguaje ensamblador.

EAX : Versión completa de 32 bits.
AX: 16 bits.
AH y AL : 8 bits respectivamente.

Registros Base. BX

Registro que trabaja por medio de índices de direcciones

EBX : Versión completa de 32 bits.
BX: 16 bits.
BH y BL : 8 bits respectivamente.

Registros Contador. CX

Contador, utilizado generalmente para contar las repeticiones de los ciclos implementados en el programa

ECX : Versión completa de 32 bits.
CX: 16 bits.
CH y CL : 8 bits respectivamente.

Registro de Datos. DX

Utilizado en la entrada y salida de datos.

EDX : Versión completa de 32 bits.
DX: 16 bits.
DH y DL : 8 bits respectivamente.

Registros Apuntadores

Apuntador de Pila

Registro de 16 bits que almacena el valor de desplazamiento de la instrucción a ejecutar

Conocido también como SP

Apuntador de Base

Registro de 16 bits que ayuda a hacer referencias a parámetros dentro de alguna subrutina.

Conocido también como BP.

Banderas

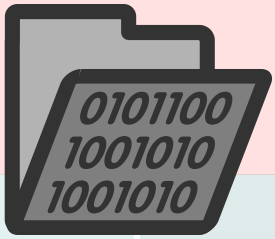
Conocidos además como banderas de estado.
Son utilizados en la mayoría de instrucciones aritméticas y lógicas como indicadores de procesos dentro de las rutinas

OVERFLOW

Indica el desbordamiento de un bit en una operación aritmética

INT FLAG

Indica la interrupción de un proceso, esta interrupción se presenta de manera externa



SIGNAL FLAG

Almacena el signo de una operación

0 = Positivo y 1= negativo

ZERO FLAG

Indica el resultado de una comparación

DIRECTION FLAG

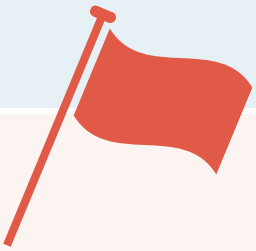
Designa la dirección para mover o en, en dado caso, comparar cadenas.

ACARREO AUXILIAR

Utilizado en aritmética especializada. Contiene un acarreo externo del tercer bit en un dato de ocho bits

PARITY FLAG

Indica la paridad o imparidad, dependiendo de su uso, de una operación.



CARRY FLAG

Contiene el acarreo resultante de operaciones aritméticas

TRAP FLAG

Inicia el proceso de depuración

Instrucciones Aritméticas

ADD

Suma sin acarreo.
Afecta a OF,SF,ZF,AF,PF,CF

DIV

División de operadores

SUB

Resta sin acarreo
Afecta a OF,SF,ZF,AF,PF,CF

INC

IIncremento del operador.
Afecta a OF,SF,ZF,AF,PF.

MUL

Multiplicación o producto
Afecta a OF, CF

DEC

Decremetno del operador.
Afecta a OF,SF,ZF,AF,PF.

Instrucciones Lógicas

AND

Operación Y entre dos operadores.
Afecta a SF,ZF,PF, OF=0,CF=0

CMP

ICompara dos operadores.
Afecta OF,SF,ZF,AF,PF,CF

OR

IOperación O entre dos operadores.
Afecta a SF,ZF,PF, OF=0,CF=0

XOR

O exclusivo entre dos operadores
Afecta a SF,ZF,PF, OF=0,CF=0

SHR

Desplazamiento derecha
Afecta a SF,ZF,AF,PF.

SHL

IDesplazamiento Izquierda
Afecta a SF,ZF,AF,PF.

Instrucciones Transferencia de Datos

LEA

Carga la dirección en registro destino.

MOV

Transfiera el origen al destino descrito en el programa

XCHG

Intercambia el contenido de los operandos escritos en la instrucción

Instrucciones Control de Flujo

CALL

"Llamar" a una subrutina en el programa.

LOOP

Transferir el control a una dirección según el valor del CX.

Instrucción utilizada para la ejecución de bucles en ensamblador

JUMP

Transferir incondicionalmente el control a una subrutina, o bien, a una dirección

Existen distintos tipos de saltos

JE

Salta si los operandos comparados son iguales.

ZF=1.

JG

Salta si el primer operando es mayor que el segundo. Con signo

SF=OF o ZF=0

JA

Salta si el primer operando es mayor que el segundo, sin signo

CF=0, ZF=0

JB

Salta si el primer operando es menor que el segundo, sin signo

CF=1

https://medium.com/nabucodonosor-editorial/saltos-en-ensamblador-x86-a94474f58ad1

Interrupciones

Las interrupciones consisten en romper el orden de una rutina y/o programa con el fin de ejecutar una "rutina de servicio". Estos son programas especiales y una vez terminan su ejecución vuelven al punto de donde fueron llamados

Algunas de las más comunes son:

- **INT 10H (Video)**
- **INT 21H(Sistema)**
- **INT 16H(Teclado)**

Si desea revisar y probar algunas de las interrupciones dentro de estos servicios o incluso probar otras diferentes a estas revisar el siguiente enlace:

http://spike.scu.edu.au/~barry/interrupts.html

Ejecución de Programa

PRIMERO DEBERÁ ENSAMBLAR EL PROGRAMA

Para esto debe de ingresar el comando /TASM seguido del nombre del programa

LUEGO DEBERÁ ENLAZAR EL PROGRAMA

Para esto debe de ingresar el comando /TLINK seguido del nombre del programa

EJECUTAR PROGRAMA

Por último, para ejecutarlo deberá escribir el nombre del programa, así como cualquier dato que el mismo pida en la linea de comandos.

PROGRAMA DE EJEMPLO

```
; Este programa despliega un hola mundo.

datos segment

  Rotulo db "Hola Mundo!$"

datos endS

pila segment stack 'stack'
   dw 256 dup(?)
pila endS


codigo segment

main: mov ax, pila
      mov ss, ax

      mov ax, datos
      mov ds, ax

      mov ah, 09h
      lea dx, rotulo

      int 21h

      mov ax, 4C00h

      int 21h

codigo ends

end main
```