

Práctica 5

Introducción al programa DEBUG

Modos de direccionamiento del procesador 80x86

Objetivo:

1. Familiarizar al alumno con las órdenes del programa DEBUG y con los conceptos de dirección lógica y registros internos del procesador 80x86.
2. Familiarizar al alumno con los diferentes modos de direccionamiento manejados por el procesador 8088/86.

Teoría:

- Resumen sobre el programa DEBUG (DOS)
- Resumen sobre modos de direccionamiento

Desarrollo:

1. Cargar el programa DEBUG en la computadora y mediante el comando **r** observe los registros internos del procesador, realice una impresión de la pantalla resultante de la ejecución de este comando y sobre la impresión indique los tres grupos de registros así como el de banderas.
2. Utilizar el programa DEBUG de DOS para ejemplificar, ejecutar y analizar cada uno de los modos de direccionamiento manejados por el procesador 8088 de Intel. Por cada modo de direccionamiento, describa mediante los comandos e instrucciones la forma de hacer el ejemplo, la forma de ejecutarlo y la forma de verificar el resultado final.
3. Escriba y ejecute en DEBUG la(s) instrucciones necesarias para:
 - a) Colocar el valor hexadecimal 4B10h en el Acumulador
 - b) Copiar el byte más significativo del Acumulador en BL
 - c) Colocar el valor decimal 31 en DH
 - d) Copiar el registro BX a CX
 - e) Copiar el registro AX al registro de Segmento de Datos
 - f) Copiar el registro CX a la dirección lógica 4B10:1F del segmento de Datos
 - g) Colocar en memoria los bytes: 1, 2, 3, 4, 5, a partir de la dirección lógica del segmento de Datos: 4B10:2A
 - h) Con la menor cantidad de instrucciones:
Colocar en el registro CL el dato almacenado en la dirección lógica del segmento de Datos 4B10:2B. En CH el dato almacenado en 4B10:2C.
En AL el dato almacenado en 4B10:2D. En AH el dato almacenado en 4B10:2E.

- i) Colocar en la dirección del segmento de Pila: $SS \times 10h + 14h$, el valor 1234h
- j) Copiar al registro DX el valor almacenado en la dirección del segmento de Pila: $SS \times 10h + 14h$
- k) Colocar en DI la palabra almacenada en la dirección efectiva 4B12E
- l) Colocar en DH el byte almacenado en la dirección efectiva 4B12A

Incluya una captura de pantalla del programa DEBUG mostrando el resultado de la ejecución de cada instrucción.

Conclusiones y Comentarios.

Bibliografía.

Programa DEBUG

Uno de los programas de utilidad más simples y usados que se proporcionaban con el sistema operativo DOS es el depurador de programas DEBUG. Aún en los sistemas más modernos como Windows XP o Windows 7, DEBUG sigue existiendo y puede ser invocado mediante la línea de comandos. Aunque lo más frecuente es que DEBUG sea utilizado por programadores de lenguaje ensamblador, ofrece varias posibilidades convenientes que son bastante fáciles de utilizar. El depurador del DOS nos permite

- Probar programas en un entorno controlado
- Cargar, visualizar o modificar cualquier archivo
- Ejecutar programas del DOS
- Llevar a cabo operaciones de lectura/escritura en sectores físicos del disco
- Crear y ensamblar programas en lenguaje ensamblador

El programa DEBUG utiliza órdenes de una sola letra las cuales muestran a continuación.

Tabla 1. Ordenes de línea del programa DEBUG

Orden	Formato
Assemble (<i>Ensamblar</i>)	A [dirección]
Compare (<i>Comparar</i>)	C intervalo o dirección
Dump (<i>Visualizar Memoria</i>)	D [intervalo] o [dirección]
Enter (<i>Modificar Memoria</i>)	E dirección [lista]
Fill (<i>Llenar Memoria</i>)	F intervalo lista
Go (<i>Ejecutar</i>)	G [=dirección][dirección [dirección..]]
Hexarithmic (<i>Hexaritmética</i>)	H valor1 valor2
Input (<i>Entrada</i>)	I dirección_puerto
Load (<i>Cargar</i>)	L [dirección][unidad][1er. sector][número]
Move (<i>Mover</i>)	M intervalo dirección
Name (<i>Nombre</i>)	N [unidad:][camino]nombre del archivo[.ext]
Output (<i>Salida</i>)	O dirección_puerto
Proceed (<i>Continuar</i>)	P [=dirección] [valor]
Quit (<i>Abandonar editor</i>)	Q
Register (<i>Registros</i>)	R [nombre_registro]
Search (<i>Buscar</i>)	S intervalo lista
Trace (<i>Trazado Paso a Paso</i>)	T [=dirección][valor]
Unassemble (<i>Desensamblar</i>)	U [dirección] o [intervalo]
Write (<i>Grabar</i>)	W [dirección][unidad][1er. sector][número]

El formato de la orden DEBUG es:

[unidad:][camino] DEBUG [d:][p][nombre archivo[.ext]] [parametro ...]

unidad es la identificación de la unidad de disco que contiene el archivo DEBUG.EXE

DEBUG es una orden externa del DOS; por tanto, el DOS tiene que cargar el archivo del disco a memoria. Si no se especifica una unidad de disco, el DOS utilizará el valor por omisión en curso.

Camino es la lista de subdirectorios que debe recorrer el DOS para hallar el archivo DEBUG.EXE. Si no se especifica un camino, el DOS utilizará el directorio de trabajo en curso.

d es la unidad de disco que contiene el archivo que va a examinar DEBUG.

p es el camino de subdirectorios que tiene que atravesar el DOS para hallar el archivo que tiene que examinar DEBUG. Si no se especifica un camino, el DOS utiliza el directorio en curso.

nombre archivo [.ext] es el nombre del archivo que tiene que examinar DEBUG.

parámetro es un parámetro de la línea de órdenes que ha de examinar DEBUG.

Después de que ha sido invocado, DEBUG copia el programa o archivo del disco a la memoria de tal modo que se puede visualizar o modificar su contenido. Al invocar DEBUG se realiza un proceso de inicialización que se describe a continuación:

DEBUG inicia el contenido de los registros de microprocesador como sigue:

- Los registros de segmento CS, DS, ES y SS reciben como valor la dirección del primer segmento que sigue al proceso DEBUG.
- El puntero de instrucciones IP, toma el valor de 100H.
- El puntero de pila (stack pointer) SP, toma como valor el final del segmento o la parte transitoria de COMMAND.COM (lo que esté más bajo).
- El resto de los registros de propósito general se pone a 0 (cero). El registro de banderas se pone en el estado siguiente: NV UP EI PL NZ NA PO NC
- Si la línea de órdenes de DEBUG contiene el nombre de un archivo, los registros de segmento DS y ES apuntarán y los registros BX y CX contendrán el tamaño del programa.

Tabla 2. Abreviatura para estado de banderas utilizada por DEBUG

Bandera	Bandera en 1	Bandera en 0
Sobreflujo (<i>Overflow</i>)	OV	NV
Dirección (<i>Direction</i>)	DN	UP
Interrupción (<i>Interrupt</i>)	EI	DI
Signo (<i>Sign</i>)	NG	PL
Cero (<i>Zero</i>)	ZR	NZ
Acarreo auxiliar (<i>Auxiliary carry</i>)	AC	NA
Paridad (<i>Parity</i>)	PE	PO
Acarreo (<i>Carry</i>)	CY	NC

```

      ↓                               ↓
34CD:1280 50 41 E8 12 02 58 83 ED-0A 1F 5B 45 8B E5 53 0B PA...X....[E..S.
34CD:1290 C0 74 03 E8 CE F4 C3 33-C0 EB EB B9 04 00 8B FC .t.....3.....
34CD:12A0 83 C7 0B E8 F1 01 C3 FD-33 C0 2B D9 74 11 7C 0E .....3.+..t..|.
34CD:12B0 87 CB 03 D9 50 B0 20 F3-AA 58 87 CB EB 01 40 88 ....P. ..X....@.
34CD:12C0 0D 4F C3 FD 50 51 57 FD-B9 09 00 8B FD 4F 32 C0 .O..PQW.....O2.
34CD:12D0 F3 AE 75 04 C6 46 00 0C-5F 59 58 C3 64 00 FD 57 ..u..F.._YX.d..W
34CD:12E0 56 83 EE 03 8B FE B9 07-00 AC 3C 09 76 0E 8A E0 V.....<.v...
34CD:12F0 D0 EC D0 EC D0 EC D0 EC-24 0F D5 0A AA E2 EA 5E .....$......^

```

Figura 1 Típica salida producida por el comando D.