



Universidad Autónoma de
Baja California

Facultad de Ciencias
Químicas e Ingeniería

Organización de las Computadoras y Lenguaje Ensamblador

Introducción al programa DEBUG

y

Modos de direccionamiento del procesador 80x86

Integrantes: Saul Ivan Flores – 01217102

Fecha: Marzo 8, 2018

Debug

Debug es un comando en DOS, OS/2 y Windows(32 bits). Se ejecuta utilizando el programa debug.exe o debug.com en version mas antiguas. El programa debug tiene varias funciones como simular un ensamblador, ensamblador, o un volcado hexadecimal habilitando a los usuario examinar contenidos de memoria interactivamente.

Debug fue desarrollado por Tim Paterson para servir su propósito en QDOS. En 1980 Tim ingreso al equipo de desarrollo de Microsoft e introdujo DOS a la compañía generando MS-DOS. DOS aunque muy util cuenta con varios limitantes como el hecho que solo puede acceder a registros de 16 bits, y el subcomando n utilizado para nombrar archivos solo puede guardar archivos en FAT 8.3.

Modos de Direccionamiento

x86 lenguaje ensamblador es una familia lenguajes ensamblador con compatibilidad reversible que provee un nivel de compatibilidad hasta el procesador intel 8008. Como todos los lenguajes de ensamblador utiliza mnemotécnicas para representar instrucciones fundamentales que la CPU en un ordenador puede entender y seguir. El lenguaje contiene diversos modos de direccionamiento.

El direccionamiento a registro consiste de transferir un byte o palabra de desde un registro hacia el registro deseado. No se permite transferir datos de un registro que tengan una cantidad distinta de bits. Existe el registro de segmento de código este immutable(no se puede modificar). No es permitido la transferencia de datos de un registro de segmento a otro

registro de segmento.

El direccionamiento inmediato mueve un dato de tamaño byte o palabra a una localidad de memoria o registro.

Direccionamiento directo transfiere un dato de tamaño byte o palabra entre un registro u una localidad de memoria del segmento, el desplazamiento contiene el valor constante '10h'

Direccionamiento de registro indirecto permite apuntar a datos almacenados en cualquier localidad de memoria del segmento de datos, de pila o del segmento extra por medio de los registros BX, BP, SI y DI.

Direccionamiento de Base más Índice es muy usado por ensambladores cuando se llaman a las funciones. Consiste al igual que el indirecto a través de registros. El contenido del registro índice indica el desplazamiento que se produce a partir de una dirección de memoria que se pasa también como argumento a la orden que utiliza este modo de direccionamiento.

Direccionamiento relativo a base permite códigos reentrantes y acceder de forma fácil y rápida a posiciones cercanas de memoria. Este modo de direccionamiento es muy usado por los ensambladores cuando se llaman a las funciones (para acceder a los parámetros almacenados en la pila).

Desarrollo

Se carga el programa debug en el ordenador y se ejecuta el comando **r** desplegando los registros del procesador. En **rojo**: Registros de propósito general, **Blanco**: Registros Punteros y de Índice, En **verde**: Registros de segmentos, **Azul**: Registros de banderas.

```
Z:\>SET BLASTER=A220 I7 D1 H5 T6

Z:\>mount x: C:/Users/SaulF/Desktop
Drive X is mounted as local directory C:/Users/SaulF/Desktop\

Z:\>X:

X:\>debug
-r
AX=0000 BX=0000 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=073F ES=073F SS=073F CS=073F IP=0100  NO UP EI PL NZ NA PO NC
073F:0100 0000          ADD     [BX+SI],AL          DS:0000=CD
```

Utilizando el programa debug se ejecutó los diversos modos de direccionamiento manejados por el procesador 8088:

Direccionamiento a Registro

Se visualizan los registros de propósito general utilizando el comando **'-r'**, se ingresa al modo ensamblador y se ingresa el comando de movimiento de registro **'mov CX, BX'**, se cierra el modo ensamblador y se ejecuta el comando con **'-t'**, para poder verificar el cambio reingresamos el comando **'r'**.

```
-r
AX=0874 BX=0000 CX=0874 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=073F ES=073F SS=073F CS=073F IP=0107  NO UP EI PL NZ NA PO NC
073F:0107 0000          ADD     [BX+SI],AL          DS:0000=CD
-a
073F:0107 mov CX,BX
073F:0109
-t
AX=0874 BX=0000 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=073F ES=073F SS=073F CS=073F IP=0109  NO UP EI PL NZ NA PO NC
073F:0109 0000          ADD     [BX+SI],AL          DS:0000=CD
-r
AX=0874 BX=0000 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=073F ES=073F SS=073F CS=073F IP=0109  NO UP EI PL NZ NA PO NC
073F:0109 0000          ADD     [BX+SI],AL          DS:0000=CD
```

Direcccionamiento Inmediato

```
-r
AX=0874 BX=0000 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=073F ES=073F SS=073F CS=073F IP=0109  NU UP EI PL NZ NA PO NC
073F:0109 0000      ADD     [BX+SI],AL      DS:0000=CD
-a
073F:0109 mov BX,07
073F:010C
-r
AX=0874 BX=0000 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=073F ES=073F SS=073F CS=073F IP=0109  NU UP EI PL NZ NA PO NC
073F:0109 BB0700      MOV     BX,0007
-t
AX=0874 BX=0007 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=073F ES=073F SS=073F CS=073F IP=010C  NU UP EI PL NZ NA PO NC
073F:010C 0000      ADD     [BX+SI],AL      DS:0007=FF
-r
AX=0874 BX=0007 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=073F ES=073F SS=073F CS=073F IP=010C  NU UP EI PL NZ NA PO NC
073F:010C 0000      ADD     [BX+SI],AL      DS:0007=FF
```

Direcccionamiento Directo

```
-r
AX=0874 BX=0007 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=073F ES=073F SS=073F CS=073F IP=010C  NU UP EI PL NZ NA PO NC
073F:010C 0000      ADD     [BX+SI],AL      DS:0007=FF
-a
073F:010C mov DL, [023]
073F:0110
-t
AX=0874 BX=0007 CX=0000 DX=00FF SP=00FD BP=0000 SI=0000 DI=0000
DS=073F ES=073F SS=073F CS=073F IP=0110  NU UP EI PL NZ NA PO NC
073F:0110 00F0      ADD     AL,DH
-r
AX=0874 BX=0007 CX=0000 DX=00FF SP=00FD BP=0000 SI=0000 DI=0000
DS=073F ES=073F SS=073F CS=073F IP=0110  NU UP EI PL NZ NA PO NC
073F:0110 00F0      ADD     AL,DH
```

Direccionamiento de Registro Indirecto

```
-r
AX=0874 BX=0007 CX=0000 DX=00FF SP=00FD BP=0000 SI=0000 DI=0000
DS=073F ES=073F SS=073F CS=073F IP=0110  NU UP EI PL NZ NA PO NC
073F:0110 00F0          ADD     AL,DH
-a
073F:0110 mov AX,[SP]
          ^ Error
073F:0110
-r
AX=0874 BX=0007 CX=0000 DX=00FF SP=00FD BP=0000 SI=0000 DI=0000
DS=073F ES=073F SS=073F CS=073F IP=0110  NU UP EI PL NZ NA PO NC
073F:0110 00F0          ADD     AL,DH
-a
073F:0110 mov AX,[SI]
073F:0112
-t
AX=20CD BX=0007 CX=0000 DX=00FF SP=00FD BP=0000 SI=0000 DI=0000
DS=073F ES=073F SS=073F CS=073F IP=0112  NU UP EI PL NZ NA PO NC
073F:0112 06          PUSH    ES
```

Direccionamiento Base más Índice

```
-a
073F:0112 mov CX, [BX+SI]
073F:0114
-r
AX=20CD BX=0007 CX=0000 DX=00FF SP=00FD BP=0000 SI=0000 DI=0000
DS=073F ES=073F SS=073F CS=073F IP=0112  NU UP EI PL NZ NA PO NC
073F:0112 8B08          MOV     CX,[BX+SI]          DS:0007=ADFF
-a
073F:0114 mov CX, [BX+SI]
073F:0116
-t
AX=20CD BX=0007 CX=ADFF DX=00FF SP=00FD BP=0000 SI=0000 DI=0000
DS=073F ES=073F SS=073F CS=073F IP=0114  NU UP EI PL NZ NA PO NC
073F:0114 8B08          MOV     CX,[BX+SI]          DS:0007=ADFF
```

Direcccionamiento Relativo a Registro

```
-r
AX=20CD BX=0007 CX=ADFF DX=00FF SP=00FD BP=0000 SI=0000 DI=0000
DS=073F ES=073F SS=073F CS=073F IP=0114  NU UP EI PL NZ NA PO NC
073F:0114 8B08          MOV     CX,[BX+SI]          DS:0007=ADFF
-mov AL,[BX+8]
^ Error
-r
AX=20CD BX=0007 CX=ADFF DX=00FF SP=00FD BP=0000 SI=0000 DI=0000
DS=073F ES=073F SS=073F CS=073F IP=0114  NU UP EI PL NZ NA PO NC
073F:0114 8B08          MOV     CX,[BX+SI]          DS:0007=ADFF
-a
073F:0116 mov AL,[BX+8]
073F:0119
-t

AX=20CD BX=0007 CX=ADFF DX=00FF SP=00FD BP=0000 SI=0000 DI=0000
DS=073F ES=073F SS=073F CS=073F IP=0116  NU UP EI PL NZ NA PO NC
073F:0116 8A4708       MOV     AL,[BX+08]          DS:000F=03
```

Direcccionamiento Relativo a Base más Índice

```
-r
AX=20CD BX=0007 CX=ADFF DX=00FF SP=00FD BP=0000 SI=0000 DI=0000
DS=073F ES=073F SS=073F CS=073F IP=0116  NU UP EI PL NZ NA PO NC
073F:0116 8A4708       MOV     AL,[BX+08]          DS:000F=03
-mov AX,[BX+SI+8]
^ Error
-r
AX=20CD BX=0007 CX=ADFF DX=00FF SP=00FD BP=0000 SI=0000 DI=0000
DS=073F ES=073F SS=073F CS=073F IP=0116  NU UP EI PL NZ NA PO NC
073F:0116 8A4708       MOV     AL,[BX+08]          DS:000F=03
-a
073F:0119 mov AX,[BX+SI+8]
073F:011C
-t

AX=2003 BX=0007 CX=ADFF DX=00FF SP=00FD BP=0000 SI=0000 DI=0000
DS=073F ES=073F SS=073F CS=073F IP=0119  NU UP EI PL NZ NA PO NC
073F:0119 8B4008       MOV     AX,[BX+SI+08]        DS:000F=A303
```

Se escribió diferentes instrucciones y se ejecutan en debug:

- a) Colocar el valor hexadecimal 4B10h en el Acumulador

```
-r
AX=0000 BX=0000 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=073F ES=073F SS=073F CS=073F IP=0100  NU UP EI PL NZ NA PO NC
073F:0100 0000          ADD     [BX+SI],AL          DS:0000=CD
-a
073F:0100 mov AX,4B10
073F:0103
-t

AX=4B10 BX=0000 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=073F ES=073F SS=073F CS=073F IP=0103  NU UP EI PL NZ NA PO NC
073F:0103 0000          ADD     [BX+SI],AL          DS:0000=CD
```

- b) Copiar el byte más significativo del Acumulador en B

```
-r
AX=4B10 BX=0000 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=073F ES=073F SS=073F CS=073F IP=0103  NU UP EI PL NZ NA PO NC
073F:0103 0000          ADD     [BX+SI],AL          DS:0000=CD
-a
073F:0103 mov BH,AH
073F:0105
-t

AX=4B10 BX=4B00 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=073F ES=073F SS=073F CS=073F IP=0105  NU UP EI PL NZ NA PO NC
073F:0105 0000          ADD     [BX+SI],AL          DS:4B00=00
```

- c) Colocar el valor decimal 31 en DH

```
-r
AX=4B10 BX=4B00 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=073F ES=073F SS=073F CS=073F IP=0105  NU UP EI PL NZ NA PO NC
073F:0105 0000          ADD     [BX+SI],AL          DS:4B00=00
-a
073F:0105 mov DH,1F
073F:0107
-t

AX=4B10 BX=4B00 CX=0000 DX=1F00 SP=00FD BP=0000 SI=0000 DI=0000
DS=073F ES=073F SS=073F CS=073F IP=0107  NU UP EI PL NZ NA PO NC
073F:0107 0000          ADD     [BX+SI],AL          DS:4B00=00
```


d) Copiar el registro BX a CX

```
-r
AX=0000 BX=0000 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=073F ES=073F SS=073F CS=073F IP=0100  NV UP EI PL NZ NA PO NC
073F:0100 0000          ADD     [BX+SI],AL          DS:0000=CD
-a
073F:0100 mov CX,BX
073F:0102
-t

AX=0000 BX=0000 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=073F ES=073F SS=073F CS=073F IP=0102  NV UP EI PL NZ NA PO NC
073F:0102 0000          ADD     [BX+SI],AL          DS:0000=CD
-
```

e) Copiar el registro AX al registro de Segmento de Datos

```
-r
AX=0000 BX=0000 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=073F ES=073F SS=073F CS=073F IP=0100  NV UP EI PL NZ NA PO NC
073F:0100 0000          ADD     [BX+SI],AL          DS:0000=CD
-a
073F:0100 mov DS,AX
073F:0102
-t

AX=0000 BX=0000 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=0000 ES=073F SS=073F CS=073F IP=0102  NV UP EI PL NZ NA PO NC
073F:0102 0000          ADD     [BX+SI],AL          DS:0000=60
-
```

f) Copiar el registro CX a la dirección lógica 4B10:1F del segmento de Datos

```

-r
AX=0000 BX=0000 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=0000 ES=073F SS=073F CS=073F IP=0102  NU UP EI PL NZ NA PO NC
073F:0102 0000      ADD     [BX+SI],AL      DS:0000=60
-a
073F:0102 mov [4B10:1F], CX
      ^ Error
073F:0102 mov [4B10], CX
073F:0106
-T
AX=0000 BX=0000 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=0000 ES=073F SS=073F CS=073F IP=0106  NU UP EI PL NZ NA PO NC
073F:0106 0000      ADD     [BX+SI],AL      DS:0000=60

```

- g) Colocar en memoria los bytes: 1, 2, 3, 4, 5, a partir de la dirección lógica del -segmento de Datos: 4B10:2A

Se utilizó el comando 'mov [2a], byte 1' para ingresar el dato a la dirección 2a, se visualiza en la imagen inferior ya que debug no contiene la función scroll.

```

-r
AX=0000 BX=0000 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=073F ES=073F SS=073F CS=073F IP=0100  NU UP EI PL NZ NA PO NC
073F:0100 0000      ADD     [BX+SI],AL      DS:0000=CD
-a
073F:0100 mov ax, 4b10
073F:0103 mov ds, ax
-t
AX=4B10 BX=0000 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=073F ES=073F SS=073F CS=073F IP=0103  NU UP EI PL NZ NA PO NC
073F:0103 8ED8      MOV     DS,AX
-t
AX=4B10 BX=0000 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=4B10 ES=073F SS=073F CS=073F IP=0105  NU UP EI PL NZ NA PO NC
073F:0105 0000      ADD     [BX+SI],AL      DS:0000=60
AX=4B10 BX=0000 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=4B10 ES=073F SS=073F CS=073F IP=010A  NU UP EI PL NZ NA PO NC
073F:010A C6062B0002 MOV     BYTE PTR [002B],02      DS:002B=60
-t
AX=4B10 BX=0000 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=4B10 ES=073F SS=073F CS=073F IP=010F  NU UP EI PL NZ NA PO NC
073F:010F C6062C0003 MOV     BYTE PTR [002C],03      DS:002C=60
-t
AX=4B10 BX=0000 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=4B10 ES=073F SS=073F CS=073F IP=0114  NU UP EI PL NZ NA PO NC
073F:0114 C6062D0004 MOV     BYTE PTR [002D],04      DS:002D=60
-t
AX=4B10 BX=0000 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=4B10 ES=073F SS=073F CS=073F IP=0119  NU UP EI PL NZ NA PO NC
073F:0119 C6062D0005 MOV     BYTE PTR [002D],05      DS:002D=04
-t
-d ds:2a
4B10:0020          01 02 03 05 00 00          .....
4B10:0030 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
4B10:0040 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
4B10:0050 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
4B10:0060 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
4B10:0070 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
4B10:0080 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
4B10:0090 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
4B10:00A0 00 00 00 00 00 00 00 00-00 00

```

- 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

```

AX=0504 BX=0000 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=4B10 ES=073F SS=073F CS=073F IP=0131 NU UP EI PL NZ NA PE NC
073F:0131 0000 ADD [BX+SI],AL DS:0000=24
-a
073F:012F mov cl, [2b]
073F:0133 mov ch, [2c]
073F:0137 mov al, [2d]
073F:013A mov ah, [2e]
073F:013E
-t

AX=04E0 BX=0000 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=4B10 ES=073F SS=073F CS=073F IP=0133 NU UP EI PL NZ NA PD NC
073F:0133 8A2E2C00 MOV CH,[002C] DS:002C=03
-t

AX=04E0 BX=0000 CX=0300 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=4B10 ES=073F SS=073F CS=073F IP=0137 NU UP EI PL NZ NA PD NC
073F:0137 A02D00 MOV AL,[002D] DS:002D=04
-t

AX=0404 BX=0000 CX=0300 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=4B10 ES=073F SS=073F CS=073F IP=013A NU UP EI PL NZ NA PD NC
073F:013A 8A262E00 MOV AH,[002E] DS:002E=05
-t

AX=0504 BX=0000 CX=0300 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=4B10 ES=073F SS=073F CS=073F IP=013E NU UP EI PL NZ NA PD NC
073F:013E 0000 ADD [BX+SI],AL DS:0000=24
-t

AX=0504 BX=0000 CX=0300 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=4B10 ES=073F SS=073F CS=073F IP=0140 NU UP EI PL NZ NA PE NC
073F:0140 0000 ADD [BX+SI],AL DS:0000=28

```

- i) Colocar en la dirección del segmento de Pila: $SS \times 10h + 14h$, el valor 1234h

```
-a
073F:0100 mov bp,14
073F:0103 mov word ptr [bp],1234
073F:0108
-t

AX=0000 BX=0000 CX=0000 DX=0000 SP=00FD BP=0014 SI=0000 DI=0000
DS=073F ES=073F SS=073F CS=073F IP=0103  NU UP EI PL NZ NA PO NC
073F:0103 C746003412  MOV     WORD PTR [BP+00],1234          SS:0014=01A3
-t

AX=0000 BX=0000 CX=0000 DX=0000 SP=00FD BP=0014 SI=0000 DI=0000
DS=073F ES=073F SS=073F CS=073F IP=0108  NU UP EI PL NZ NA PO NC
073F:0108 06          PUSH     ES
-d ds:14
073F:0010
073F:0020 FF FF FF FF FF FF FF FF FF FF FF 00 00 00 00 4.....
073F:0030 00 00 14 00 18 00 3F 07-FF FF FF FF 00 00 00 00 .....?.....
073F:0040 05 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
073F:0050 CD 21 CB 00 00 00 00 00-00 00 00 00 00 20 20 20 .?.....
073F:0060 20 20 20 20 20 20 20 20-00 00 00 00 00 20 20 20 .....
073F:0070 20 20 20 20 20 20 20 20-00 00 00 00 00 00 00 00 .....
073F:0080 00 0D 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
073F:0090 00 00 00 00
```

- j) Copiar al registro DX el valor almacenado en la dirección del segmento de Pila: $SS \times 10h + 14$

```
-a
073F:0108 mov dx,[14]
073F:010C
-t

AX=0000 BX=0000 CX=0000 DX=1234 SP=00FD BP=0014 SI=0000 DI=0000
DS=073F ES=073F SS=073F CS=073F IP=010C  NU UP EI PL NZ NA PO NC
073F:010C C606AEFE00  MOV     BYTE PTR [FEAE],00          DS:FEAE=00
```

k) Colocar en DI la palabra almacenada en la dirección efectiva 4B12E

```

-r
AX=0000 BX=0000 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=073F ES=073F SS=073F CS=073F IP=0100  NU UP EI PL NZ NA PO NC
073F:0100 BB1400          MOV     BX,0014
-a
073F:0100 mov bx, 14
073F:0103 mov di, word ptr[bx]
073F:0105
-t

AX=0000 BX=0014 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=073F ES=073F SS=073F CS=073F IP=0103  NU UP EI PL NZ NA PO NC
073F:0103 8B3F          MOV     DI,[BX]          DS:0014=01A3
-t

AX=0000 BX=0014 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=01A3
DS=073F ES=073F SS=073F CS=073F IP=0105  NU UP EI PL NZ NA PO NC
073F:0105 0000          ADD     [BX+SI],AL          DS:0014=A3
+d ds:14
073F:0010          A3 01 92 01-01 01 01 00 02 FF FF FF .....
073F:0020 FF FF FF FF FF FF FF FF FF FF 00 00 00 00 .....
073F:0030 00 00 14 00 18 00 3F 07-FF FF FF FF 00 00 00 00 .....?.....
073F:0040 05 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
073F:0050 CD 21 CB 00 00 00 00 00-00 00 00 00 00 20 20 20 .!.....
073F:0060 20 20 20 20 20 20 20 20-00 00 00 00 00 20 20 20 .....
073F:0070 20 20 20 20 20 20 20 20-00 00 00 00 00 00 00 00 .....
073F:0080 00 0D 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
073F:0090 00 00 00 00 .....
-r
AX=0000 BX=0014 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=01A3
DS=073F ES=073F SS=073F CS=073F IP=0105  NU UP EI PL NZ NA PO NC
073F:0105 0000          ADD     [BX+SI],AL          DS:0014=A3

```

l) Colocar en DH el byte almacenado en la dirección efectiva 4B12A

```
-r
AX=0000 BX=0014 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=01A3
DS=073F ES=073F SS=073F CS=073F IP=0105  NU UP EI PL NZ NA PO NC
073F:0105 0000          ADD     [BX+SI],AL          DS:0014=A3
-a
073F:0105 mov ax, 4b12
073F:0108
-t

AX=4B12 BX=0014 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=01A3
DS=073F ES=073F SS=073F CS=073F IP=0108  NU UP EI PL NZ NA PO NC
073F:0108 0000          ADD     [BX+SI],AL          DS:0014=A3
-a
073F:0108 mov ds, ax
073F:010A
-t

AX=4B12 BX=0014 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=01A3
DS=4B12 ES=073F SS=073F CS=073F IP=010A  NU UP EI PL NZ NA PO NC
073F:010A 0000          ADD     [BX+SI],AL          DS:0014=00
-a
073F:0114 mov bx,a
073F:0117 mov dh,[bx]
073F:0119
-t

AX=4B12 BX=000A CX=0000 DX=1200 SP=00FD BP=0000 SI=0000 DI=01A3
DS=4B12 ES=073F SS=073F CS=073F IP=0117  NU UP EI PL NZ NA PO NC
073F:0117 8A37          MOV     DH,[BX]          DS:000A=12
```

Conclusiones

Se da conocer como el lenguaje ensamblador direcciona datos a localidades de memorias. Esto nos da las habilidades de editar y copiar direcciones o datos. Es importante conocer todas las maneras de direccionamiento ya que nos evitamos con problemas a la hora de estar programando. La herramienta debug es muy útil para probar los comandos antes de meterlos a un programa oficial, ya que tenemos un entorno que no lastimara ningún dispositivo que utilice el programa a ejecutar. Para poder utilizar la herramienta debug se utilizó el programa DOSBOX esto nos permite emular un entorno de 32 bits y ejecuta el programa debug.exe sin problemas una vez que se montó el directorio virtual.

- Flores Torres Coto Saul Ivan