

LẬP TRÌNH HỆ THỐNG.

1. **Môi trường sử dụng Debug.**
2. **Kiến thức bổ trợ.**
3. **Một số lệnh và cấu trúc điều khiển trong ASM.**

1. Môi trường sử dụng Debug.

- Môi trường sử dụng Debug.

Windows x86

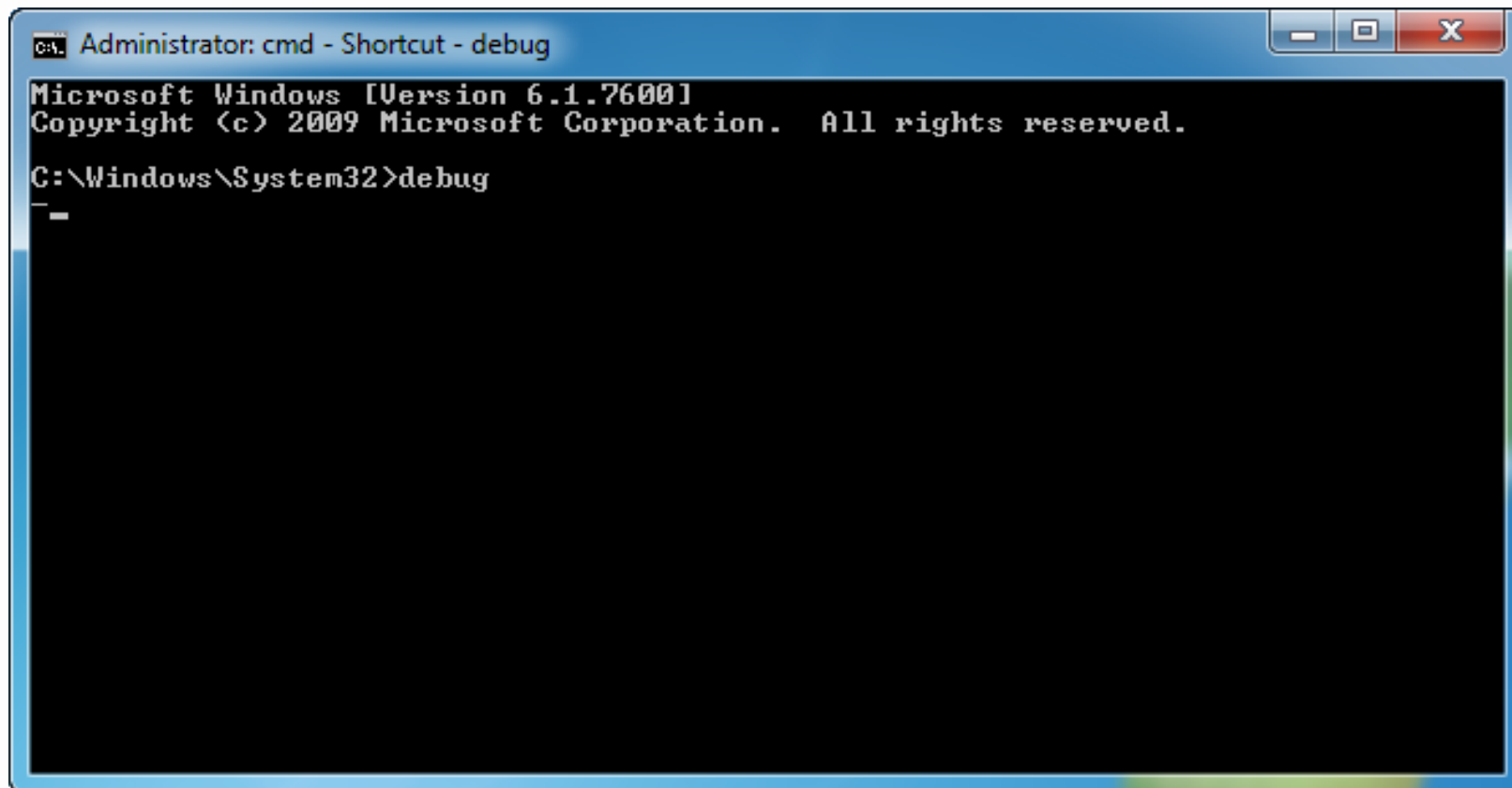
Windows x86 trong máy ảo (Virtual Machine).

1. Môi trường sử dụng Debug.

- Debug.

Khởi động CMD.

Debug.



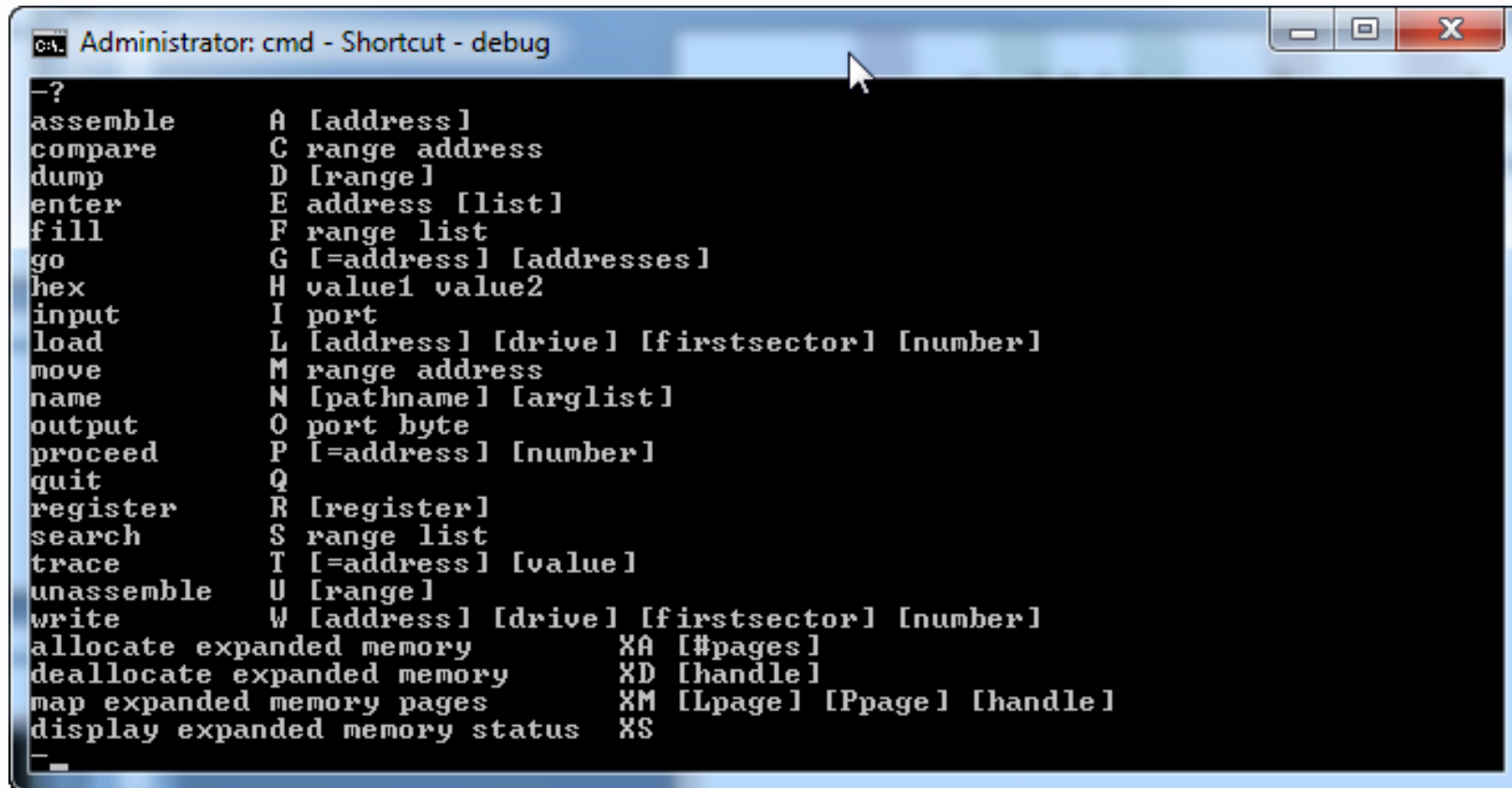
```
Administrator: cmd - Shortcut - debug
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Windows\System32>debug
_
```

1. Môi trường sử dụng Debug.

Lệnh trợ giúp :

?



```
Administrator: cmd - Shortcut - debug

-?
assemble      A [address]
compare       C range address
dump          D [range]
enter         E address [list]
fill          F range list
go            G [=address] [addresses]
hex           H value1 value2
input         I port
load          L [address] [drive] [firstsector] [number]
move          M range address
name          N [pathname] [arglist]
output        O port byte
proceed       P [=address] [number]
quit          Q
register       R [register]
search        S range list
trace         T [=address] [value]
unassemble    U [range]
write         W [address] [drive] [firstsector] [number]
allocate expanded memory  XA [#pages]
deallocate expanded memory XD [handle]
map expanded memory pages XM [Lpage] [Ppage] [handle]
display expanded memory status XS
-
```

1. Môi trường sử dụng Debug.

Lệnh assembler	: a [address]
Lệnh dump	: d [range]
Lệnh enter	: e address [list]
Lệnh go	: g [=address] [address]
Lệnh hex	: h value1 value2
Lệnh load	: l [address] [drive] [firstsector] [number]
Lệnh name	: n [pthname] [arglist]
Lệnh proceed	: p [address] [number]
Lệnh quit	: q
Lệnh register	: r [register]
Lệnh unassembler	: u [range]
Lệnh write	: w [address] [drive] [firstsector] [number]

1. Môi trường sử dụng Debug.

Xem nội dung bộ nhớ :

d 0:0

```
-d 0:0
0000:0000  68 10 A7 00 8B 01 70 00-16 00 A0 03 8B 01 70 00  h.....p.....p.
0000:0010  8B 01 70 00 CC 06 0E 02-53 07 0E 02 FF 03 0E 02  ..p.....S.....
0000:0020  59 07 0E 02 0A 04 0E 02-3A 00 A0 03 54 00 A0 03  y.....:...T...
0000:0030  6E 00 A0 03 88 00 A0 03-A2 00 A0 03 FF 03 0E 02  n.....
0000:0040  BD 08 0E 02 E2 09 0E 02-E8 09 0E 02 5D 04 0E 02  .....l...
0000:0050  EE 09 0E 02 0D 02 E1 02-00 0A 0E 02 8B 05 0E 02  .....
0000:0060  4A 0C 0E 02 50 0C 0E 02-5B 0C 0E 02 C0 06 0E 02  J...P...l.....
0000:0070  C0 06 0E 02 00 00 00 00-37 05 0E 02 65 6E 00 C0  .....7...en..
```

1. Môi trường sử dụng Debug.

Soạn chương trình Assembly :
Không có data.

a 100

```
-a 100  
1390:0100 mov ah,2  
1390:0102 mov dl,41  
1390:0104 int 21  
1390:0106 int 20  
1390:0108  
_
```

Có data.

a 100

```
-a 100  
1390:0100 mov ah,9  
1390:0102 mov dx,200  
1390:0105 int 21  
1390:0107 int 20  
1390:0109  
-e 200 "Chương trình có data$"
```

1. Môi trường sử dụng Debug.

Thực thi từng dòng lệnh :

R ip

... nhập 100

P

...

1. Môi trường sử dụng Debug.

Thực thi toàn bộ CT :

R ip

... nhập 100

g

1. Môi trường sử dụng Debug.

Xem lại CT :

U 100

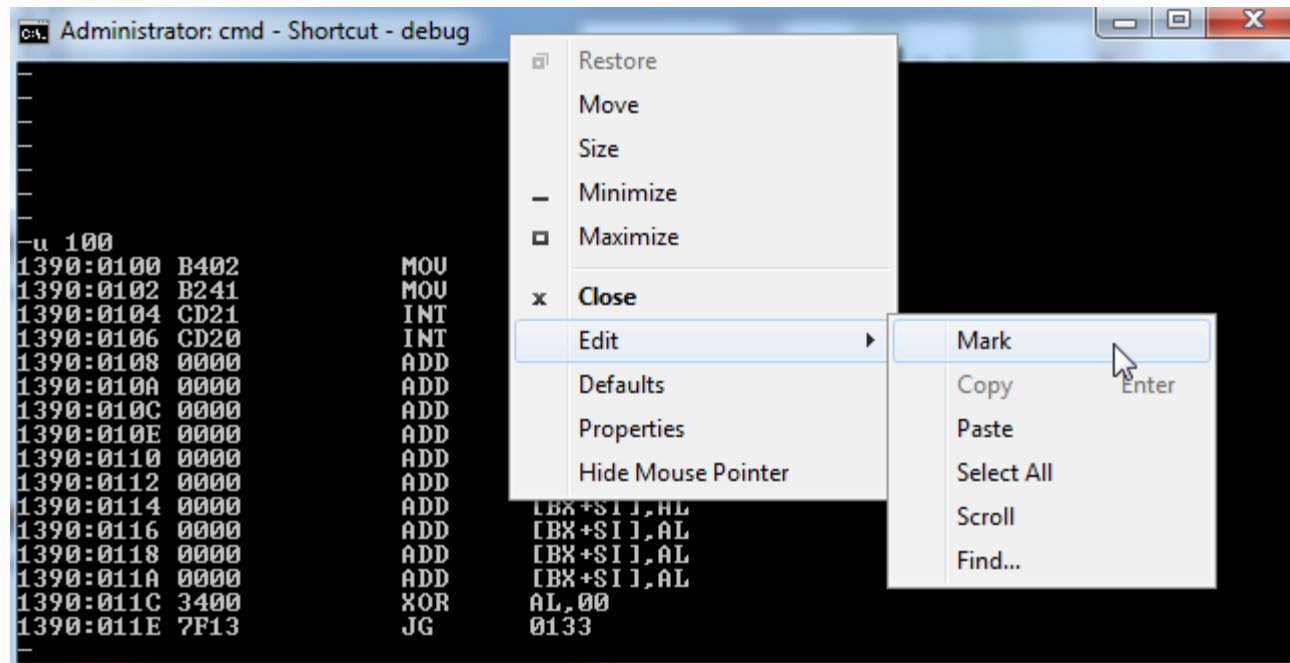
```
-u 100
1390:0100 B402      MOV     AH,02
1390:0102 B241      MOV     DL,41
1390:0104 CD21      INT     21
1390:0106 CD20      INT     20
1390:0108 0000      ADD     [BX+SI],AL
1390:010A 0000      ADD     [BX+SI],AL
1390:010C 0000      ADD     [BX+SI],AL
1390:010E 0000      ADD     [BX+SI],AL
```

1. Môi trường sử dụng Debug.

Lưu code CT dạng text :

Thanh tiêu đề/Edit/Mark

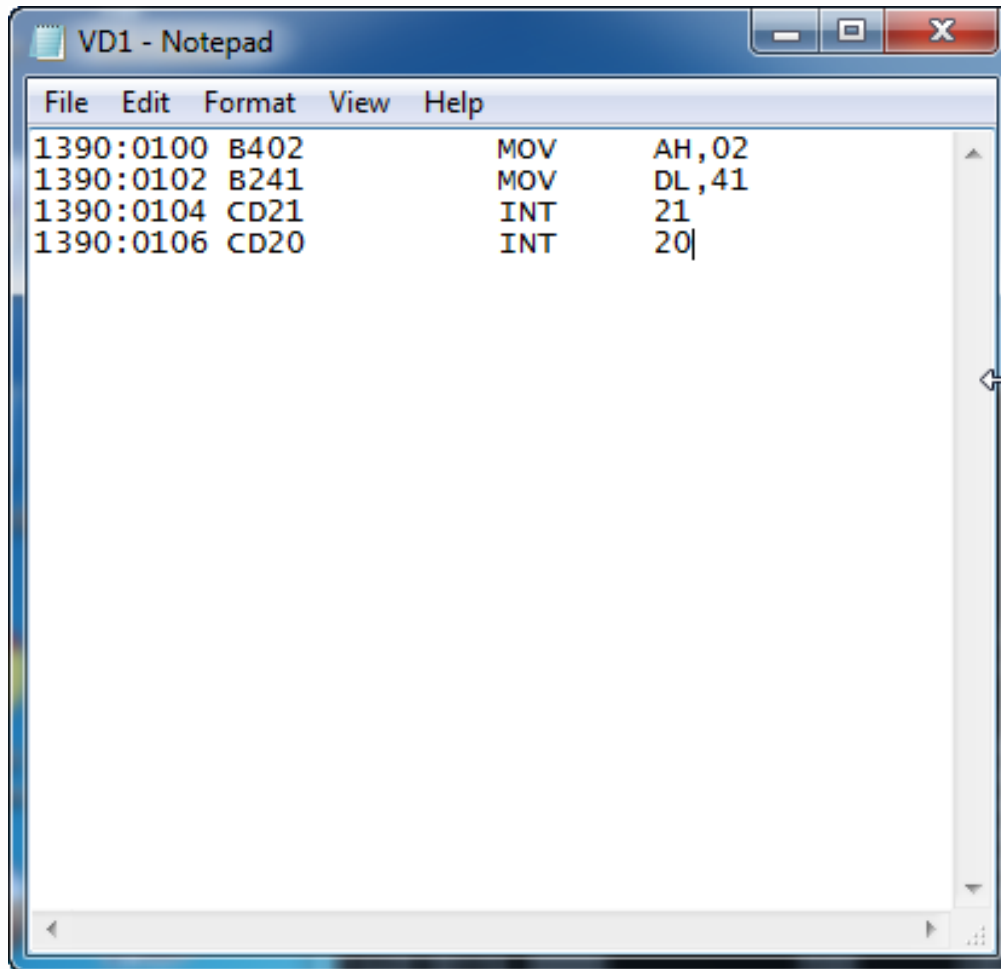
Chọn code + Enter



Address	Hex	Instruction	Comment
1390:0100	B402	MOV	AH, 02
1390:0102	B241	MOV	DL, 41
1390:0104	CD21	INT	21
1390:0106	CD20	INT	20
1390:0108	0000	ADD	[BX+SI], AL

1. Môi trường sử dụng Debug.

Mở file text + Paste



The image shows a Notepad window titled "VD1 - Notepad" with a menu bar (File, Edit, Format, View, Help) and a text area containing assembly code. The code is as follows:

Address	Hex	Op	Operand
1390:0100	B402	MOV	AH,02
1390:0102	B241	MOV	DL,41
1390:0104	CD21	INT	21
1390:0106	CD20	INT	20

1. Môi trường sử dụng Debug.

Lưu code CT dạng .com :

Tính độ dài CT : h address1(cuối) address2(đầu).

Gán độ dài CT cho thanh ghi cx : r cx (nhập độ dài CT)

Khai báo tên CT : n <ten CT>.com

Ghi CT : w

1. Môi trường sử dụng Debug.

Nạp CT dạng .com :

Khai báo tên CT : n <ten CT>.com

Nạp CT : l 100

Xem code CT : u 100

2. Kiến thức bổ trợ.

Interrupt :

INT 10h

INT 13h

INT 21h

AH = 1, 2, 8, 9, 0ah

INT 33h

Thanh ghi :

AX,BX,CX,DX,AH,AL,BH,BL,CH,CL,DH,DL

3. Một số lệnh và cấu trúc điều khiển trong ASM.

Một số lệnh asm16 :

ADD

SUB

INC

MOV

CMP

JMP

JZ, JNZ

JE, JNE

LOOP

...

3. Một số lệnh và cấu trúc điều khiển trong ASM.

Một số cấu trúc điều khiển :

IF ...

FOR ...

DO WHILE ...

3. Một số lệnh và cấu trúc điều khiển trong ASM.

IF ...

cmp a,b

jnz <đ/c1> (jz, je, jne)

...

<đ/c1>

...

```
1390:0100 B401      MOV     AH,01
1390:0102 CD21      INT     21
1390:0104 3C61      CMP     AL,61
1390:0106 7506      JNZ     010E
1390:0108 88C2      MOV     DL,AL
1390:010A B402      MOV     AH,02
1390:010C CD21      INT     21
1390:010E CD20      INT     20
```

3. Một số lệnh và cấu trúc điều khiển trong ASM.

FOR ...

a=<gtd>

cx=<so lan lap>

...

Loop <...>

...

```
1390:0100 B230      MOV     DL,30
1390:0102 B90A00    MOV     CX,000A
1390:0105 B402      MOV     AH,02
1390:0107 CD21      INT     21
1390:0109 FEC2      INC     DL
1390:010B E2F8      LOOP    0105
1390:010D CD20      INT     20
```

3. Một số lệnh và cấu trúc điều khiển trong ASM.

DO WHILE ...

<đ/c2>

...

cmp a,b

jz <đ/c1> (jnz, je, jne)

...

jmp <đ/c2>

<đ/c1>

...

1390:0100	B401	MOU	AH,01
1390:0102	CD21	INT	21
1390:0104	3C1B	CMP	AL,1B
1390:0106	7408	JZ	0110
1390:0108	88C2	MOU	DL,AL
1390:010A	B402	MOU	AH,02
1390:010C	CD21	INT	21
1390:010E	EBF0	JMP	0100
1390:0110	CD20	INT	20