

## 汇编实验 2 伪指令 及 顺序结构程序设计

### 实验目的:

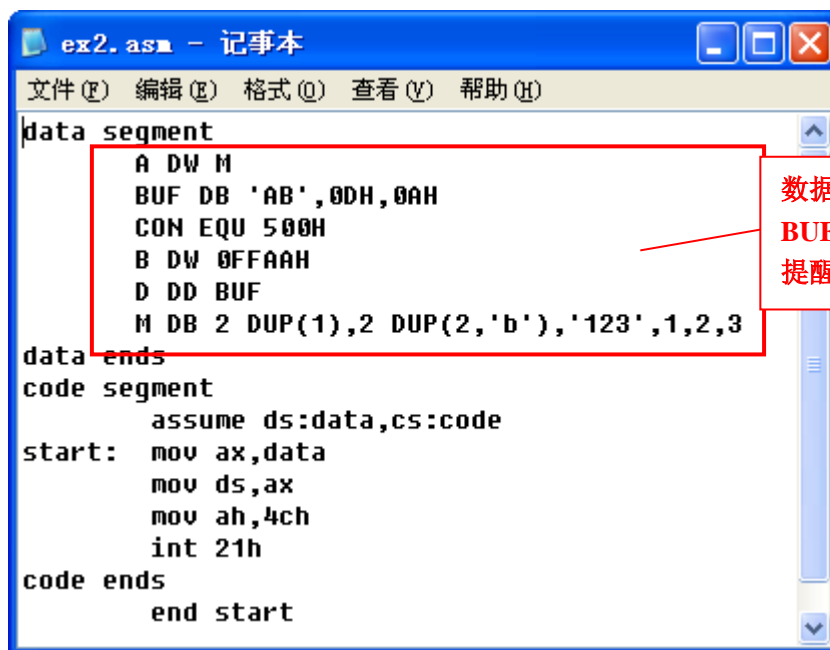
- (1) 掌握汇编语言程序上机过程。
- (2) 掌握伪指令 DB/DW, EQU, 属性操作符 SEG, LENGTH, OFFSET, TYPE, SIZE 以及 PTR 的用法。
- (3) 掌握顺序结构程序调试的方法。

### 实验内容:

[1] 伪指令部分: 验证数据段中的数据在内存单元中的存放, 画出数据段中内存单元的地址和内容示意图

操作步骤:

1. 输入源程序: `EDIT c:\hb\ex2.asm`



```
ex2.asm - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

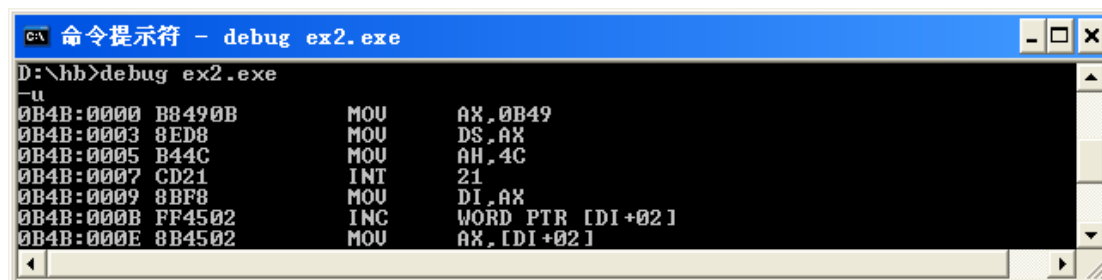
data segment
    A DW M
    BUF DB 'AB', 0DH, 0AH
    CON EQU 500H
    B DW 0FFAAH
    D DD BUF
    M DB 2 DUP(1), 2 DUP(2, 'b'), '123', 1, 2, 3
data ends
code segment
    assume ds:data, cs:code
start: mov ax, data
       mov ds, ax
       mov ah, 4ch
       int 21h
code ends
       end start
```

数据段中定义了变量 A, BUF..., 提醒: CON 是常量

在 EDIT 编辑环境 File 菜单下选择 Save 存盘, 选择 Exit 退出编辑环境

2. 汇编源程序: [masm ex2.asm](#)(具体步骤略)
3. 连接目标程序: [link ex2.obj](#)(具体步骤略)
4. 用 DEBUG 调试并查看运行结果 [DEBUG ex2.EXE](#)

① 先用 U 命令进行反汇编, 看到源程序在内存中的存储



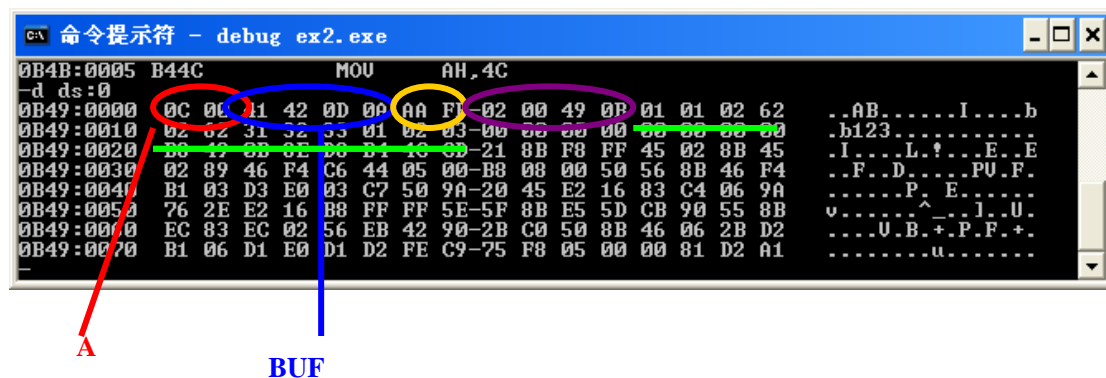
```
命令提示符 - debug ex2.exe
D:\hb>debug ex2.exe
-u
0B4B:0000 B8490B      MOV     AX, 0B49
0B4B:0003 8ED8             MOV     DS, AX
0B4B:0005 B44C             MOV     AH, 4C
0B4B:0007 CD21             INT     21
0B4B:0009 8BF8             MOV     DI, AX
0B4B:000B FF4502           INC     WORD PTR [DI+02]
0B4B:000E 8B4502           MOV     AX, [DI+02]
```

② T (单步执行一条指令。)

③ T (单步执行一条指令。)

④ D DS:0 (显示 DS 段, 偏移地址为 0000 单元开始的 128 个存储单元的内容。)

用 D 命令观看数据段的内容

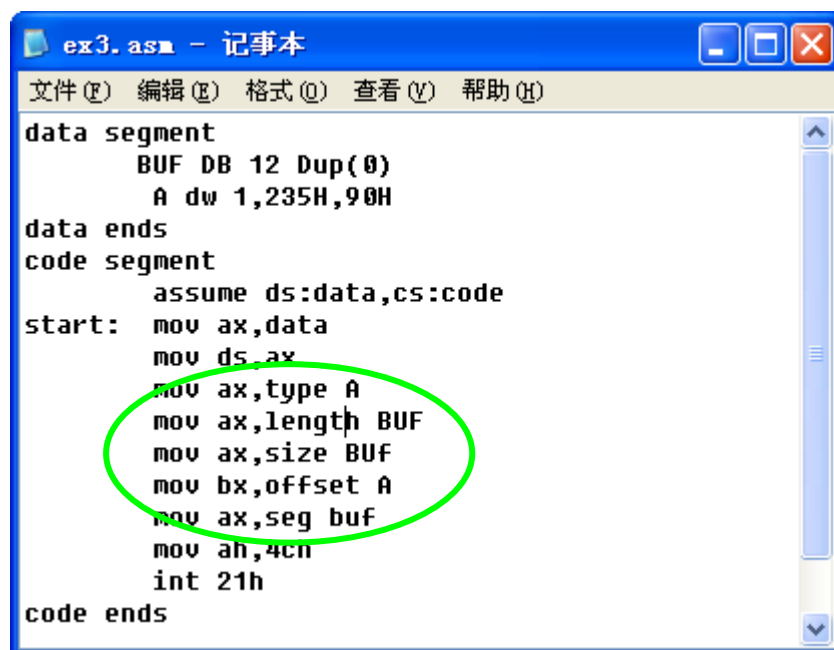


**观看结果** (用 Debug 中的 D 命令查看内存空间, 观察用 DB、DW 定义的变量在内存单元的存放, 观察 EQU 定义的表达式是否会被分配内存单元?) 画出内存示意图

## [2] 验证、掌握属性操作符 SEG, LENGTH, OFFSET, TYPE, SIZE 的用法

操作步骤:

1. 输入源程序: EDIT c:\hb\ex3.asm



2. 汇编源程序: masm ex2.asm (具体步骤略)
3. 连接目标程序: link ex2.obj (具体步骤略)
4. 用 DEBUG 调试并查看运行结果

```
命令提示符 - debug ex3.exe
D:\hb>debug ex3.exe
-u
0B4B:0000 B8490B MOU AX, 0B49
0B4B:0003 8ED8 MOU DS, AX
0B4B:0005 B80200 MOU AX, 0002
0B4B:0008 B80C00 MOU AX, 000C
0B4B:000B B80C00 MOU AX, 000C
0B4B:000E B80C00 MOU BX, 000C
0B4B:0011 B8490B MOU AX, 0B49
0B4B:0014 B44C MOU AH, 4C
0B4B:0016 CD21 INT 21
0B4B:0018 E885C2 CALL C2A0
0B4B:001B 83C404 ADD SP, +04
0B4B:001E 050C00 ADD AX, 000C
```

对照源程序中的内容，**掌握**属性操作符 SEG, LENGTH, OFFSET, TYPE, SIZE 的意义，经过汇编后形成立即数

### [3] 调试查表程序：

内存中自 TABLE 开始的七个单元中连续存放着自然数 0~6 的立方值(立方值表)，任一个数 X ( $0 \leq X \leq 6$ ) 在 FIRST 单元中，要求查表找出 X 的立方值，存 SECOND 单元。

### 操作步骤：

1. 输入源程序: **EDIT c:\hb\cb.asm** (注意: 下图文件名和源程序中有 BUG!)

```
cb.asm.txt - 记事本
文件(F) 编辑(E) 格式(O) 帮助(H)
mystack segment stack
    dw 20h dup (?)
mystack ends
data segment
    TABLE DW 0, 1, 6, 27, 64, 125, 216
    FIRST DB 3
    SECOND DB ?
data ends
code segment
    assume cs:code, ds:data, ss:mystack
start: mov ax, data
    mov ds, ax
    MOV BX, OFFSET TABLE; lea bx, TABLE
    MOV AH, 0
    MOV AL, FIRST
    ADD BX, AX
    MOV AL, [BX]
    MOV SECOND, AL
    mov ah, 4ch
    int 21h
code ends
end start
```

在 EDIT 编辑环境 File 菜单下选择 Save 存盘，选择 Exit 退出编辑环境

## 2. 汇编源程序

C:\MASM CB✓

Microsoft (R) Macro Assembler Version 5.10

Copyright (C) Microsoft Corp 1981,1988 All rights reserved.

Object filename [CB.OBJ]: ✓

Source listing [NUL.LST]: ✓

Cross-reference [NUL.CRF]: ✓

51394+389950 Bytes symbol space free

0 Warning Errors

0 Severe Errors

C:\>\_

## 3. 连接目标程序

C:\>LINK CB✓

Microsoft (R) Overlay Linker Version 3.64

Copyright (C) Microsoft Corp 1983-1988. All rights reserved.

Object Modules [.OBJ]: ✓

Run File [NUL.MAP]: ✓

Libraries [.LIB]: ✓



```
命令提示符
D:\hb>masm cb.asm
Microsoft (R) Macro Assembler Version 5.10
Copyright (C) Microsoft Corp 1981, 1988. All rights reserved.
Object filename [cb.OBJ]:
Source listing [NUL.LST]:
Cross-reference [NUL.CRF]:

48396 + 446063 Bytes symbol space free
0 Warning Errors
0 Severe Errors
D:\hb>link cb.obj
Microsoft (R) Segmented-Executable Linker Version 5.01.21
Copyright (C) Microsoft Corp 1984-1988. All rights reserved.
Run File [CB.EXE]:
List File [NUL.MAP]:
Libraries [.LIB]:
Definitions File [NUL.DEF]:
D:\hb>
```

显示 0 个错误才能进行下一步 LINK 操作

## 4. 执行程序

C:\>CB✓

没有任何结果输出，原因是程序中没有安排显示输出，结果保存在内存单元中。

## 5. 用 DEBUG 调试并查看运行结果

C:\>DEBUG CB.EXE

先用 U 命令进行反汇编，看到源程序在内存中的存储

```

命令提示符 - debug cb.exe
Microsoft (R) Segmented-Executable Linker Version 5.01.21
Copyright (C) Microsoft Corp 1984-1988. All rights reserved.

Run File [CB.EXE]:
List File [NUL.MAP]:
Libraries [LIB1]:
Definitions File [NUL.DEF]:

D:\hb>debug cb.exe
-u
0BF0:0000 B8EF0B      MOV     AX,0BEF
0BF0:0003 8ED8      MOV     DS,AX
0BF0:0005 B80000      MOV     BX,0000
0BF0:0008 B400      MOV     AH,00
0BF0:000A A00E00      MOV     AL,[000E]
0BF0:000D 03D8      ADD     BX,AX
0BF0:000F 8A07      MOV     AL,[BX]
0BF0:0011 A20F00      MOV     [000F],AL
0BF0:0014 B44C      MOV     AH,4C
0BF0:0016 CD21      INT     21
0BF0:0018 EB0B      JMP     0025
0BF0:001A B86A00      MOV     AX,006A
0BF0:001D 50      PUSH    AX
0BF0:001E B8B40D      MOV     AX,0DB4

```

T (单步执行一条指令。)

AX=0BEF BX=0000 CX=0068 DX=0000 SP=0040 BP=0000 SI=0000 DI=0000  
DS=0BDB ES=0BDB SS=0BEB CS=0BF0 IP=0003 NV UP EI PL NZ NA PO NC  
0BF0:0003 8ED8 MOV DS,AX

T (单步执行一条指令。)

AX=0BEF BX=0000 CX=0068 DX=0000 SP=0040 BP=0000 SI=0000 DI=0000  
DS=0BEF ES=0BDB SS=0BEB CS=0BF0 IP=0005 NV UP EI PL NZ NA PO NC  
0BF0:0005 B80000 MOV BX,0000

D DS:0 (显示 DS 段, 偏移地址为 0000 单元的内容。)

用 D 命令 观看数据段的内容

```

命令提示符 - debug cb.exe
-d ds:0
0BEF:0000 00 01 08 1B 40 7D D9 03 00 00 00 00 00 00 00 .....0>.....
0BEF:0010 B8 EF 0B 8E D8 D8 00 00 B4 00 A0 07 00 03 D8 8A .....L!...j.P.
0BEF:0020 07 02 07 00 B4 4C CD 21 EB 0B B8 6A 00 50 B8 B4 .....P.!...z/<.u...8>
0BEF:0030 0D 50 38 21 03 E8 25 2F 3C 2C 75 0A FF 06 38 29 .....+F..F.9v
0BEF:0040 E8 FF 1F EB 06 90 8B C6 28 46 EC 89 46 F0 39 76 .....w...+F.;F.s...3.
0BEF:0050 EC 77 0A 8B C6 2B 46 EC 31 46 F0 73 09 B8 33 00 .....P.t...3.8>.F..F..
0BEF:0060 50 E8 74 03 EB 33 A1 38 29 89 46 FE 8B 46 EC 03 .....8>.v.....F..8>
0BEF:0070 C7 A3 38 29 FF 76 F0 E8 04 FB 8B 46 FE A3 38 29

```

数据段中定义的 TABLE 立方表      FIRST 的内容      预留的存放结果的单元

T (继续单步运行若干次。)

直到程序运行到 MOV AH, 4CH 为止

或者 G 0014,

用 G 命令一次执行多条指令直到运行到断点 0014H (即指令 MOV AH, 4CH 为止)

D DS:0 (显示 DS 段, 偏移地址为 0000 单元的内容。)

```
命令提示符 - debug cb.exe
-d ds:0
0BEF:0000 00 01 08 1B 40 7D D8 03-1B 00 00 00 00 00 00 00 .....0>.....
0BEF:0010 B8 EF 0B 8E D8 BB 00 00-84 00 A0 07 00 03 D8 8A .....
0BEF:0020 07 A2 08 00 B4 4C CD 21-EB 0B 88 6A 00 50 B8 B4 .....L-!...j-P..
0BEF:0030 0D 50 E8 21 03 E8 25 2F-3C 2C 75 0A FF 06 38 29 .P.!...z/<.u...8>
0BEF:0040 E8 FF 1F EB 06 90 8B C6-2B 46 EC 82 46 F0 39 76 .....+F..F.9v
0BEF:0050 EC 77 0A 8B C6 2B 46 EC-3B 46 F0 73 09 B8 33 00 .w...+F.;F.s..3.
0BEF:0060 50 E8 74 03 EB 33 A1 38-29 89 46 FE 8B 46 EC 03 P.t..3.8>.F..F..
0BEF:0070 C7 A3 38 29 FF 76 F0 E8-04 FB 8B 46 FE A3 38 29 ..8>.v.....F..8>
```

程序结束，该单元内容就是结果（查出 3 对应的立方值为 1BH）