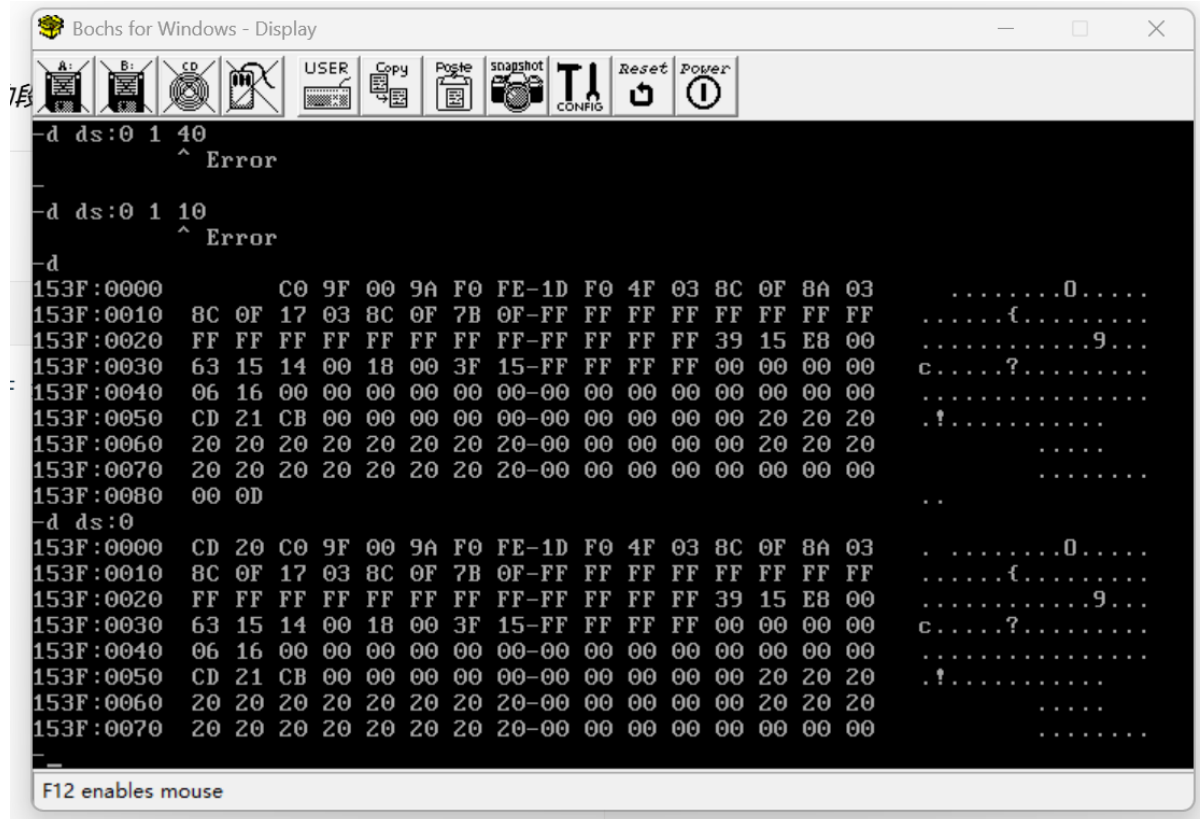
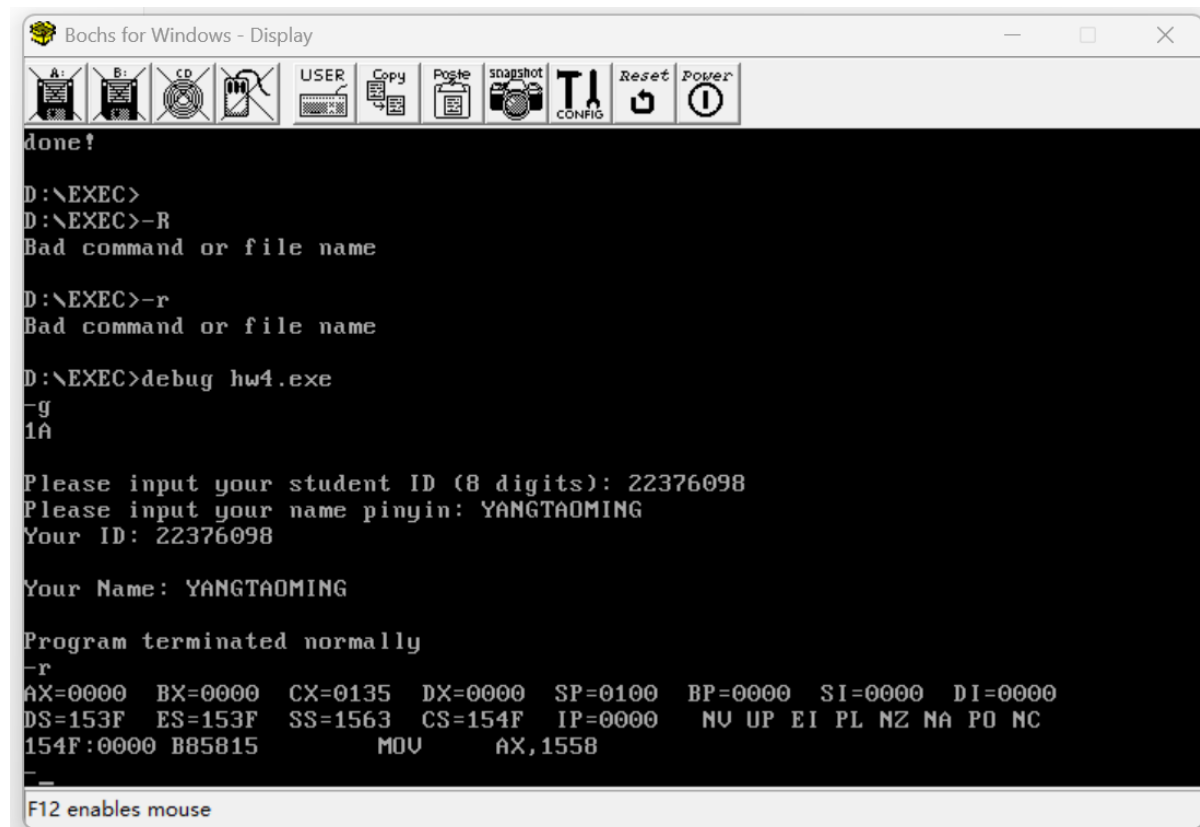


3.数据段截图如下:



DT1 占用 4 字节 (偏移 0000~0003)
DT2 占用 8 字节 (偏移 0004~000B)
DT3 存储 DT2 的偏移地址: 153F:0004

DT4 存储 DT2 长度：8 字节

4. 回答问题如下：

COUNT 是否在数据段中有对应的存储单元？

答：没有。COUNT EQU \$-DT1 是等值定义 (EQU)，在汇编时被替换为常数 4，不分配内存空间。

DT1 和 DT2 有何不同，是什么原因造成这种不同？

答：DT1 DB 1,2,'AB' → 定义 4 个字节：01H, 02H, 41H, 42H

DT2 DW 1,2,'AB' → 定义 4 个字 (8 字节)：0001H, 0002H, 0041H, 0042H 原因：DB 定义字节，DW 定义字 (2 字节)，字符 'AB' 在 DW 中被扩展为字。

DT4 的值是多少，有何含义？如果自己计算，如何计算？

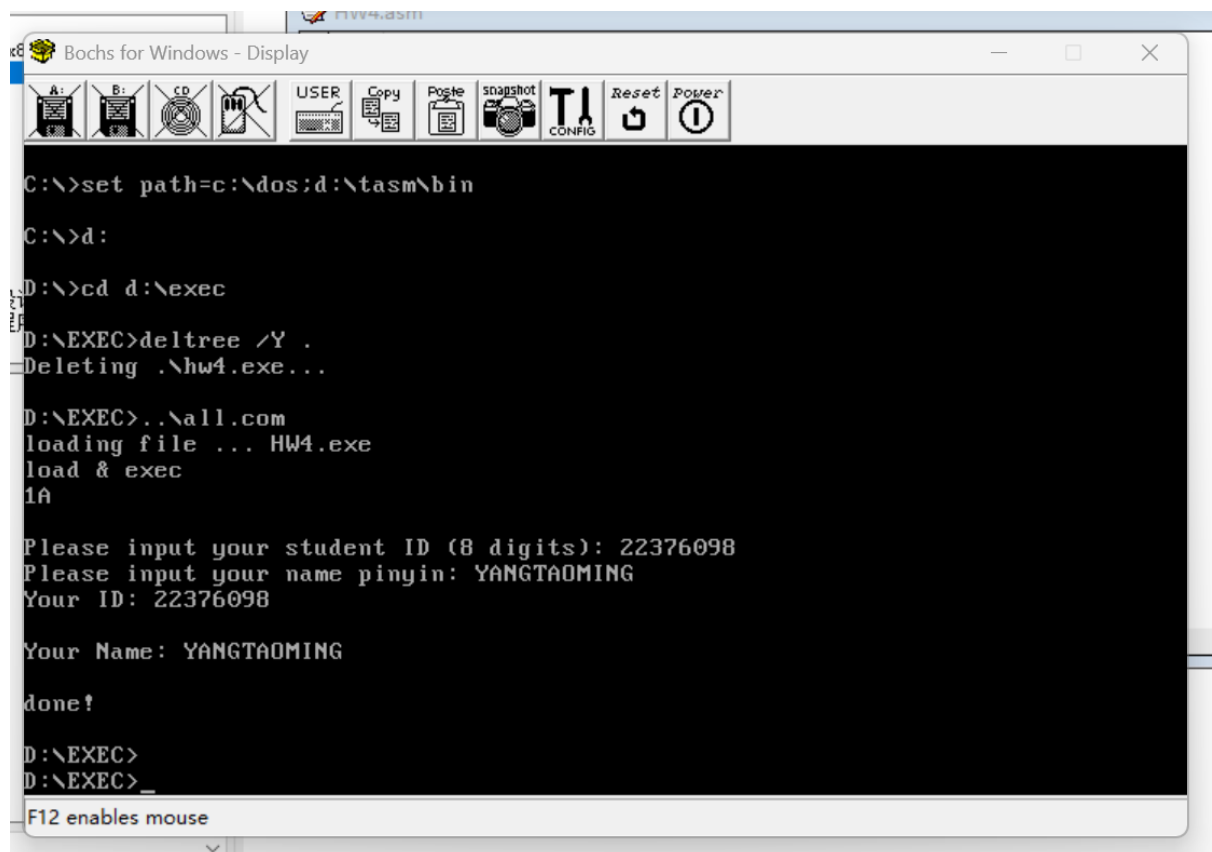
答：值：8 含义：DT2 所占的内存字节数

可以用以下程序计算：

```
DT2      DW  1,2,'AB'
```

```
DT4      DW  $-DT2
```

1. A, 学号姓名，输入输出截图如下



```
C:\>set path=c:\dos;d:\tasm\bin

C:\>d:

D:\>cd d:\nexec
D:\EXEC>deltree /Y .
Deleting .\hw4.exe...

D:\EXEC>..\all.com
loading file ... HW4.exe
load & exec
1A

Please input your student ID (8 digits): 22376098
Please input your name pinyin: YANGTAOMING
Your ID: 22376098

Your Name: YANGTAOMING

done!

D:\EXEC>
D:\EXEC>
```

三，思考题

返回 DOS 语句的作用是什么？如果缺失会有什么后果？

答：MOV AX,4C00H / INT 21H 是标准退出程序方式。

缺失后果：程序结束但未正确返回 DOS，可能导致：挂起系统，内存泄漏，后续程序无法正常运行

在定义了多个逻辑段的程序中，同时有效的逻辑段最多有几个？

答：最多 3 个：代码段 (CS)，数据段 (DS)，堆栈段 (SS)

在程序中，哪一类逻辑段必不可少？

答：代码段必不可少，没有代码段，程序无法执行。

下面是我的源程序：

```
.MODEL SMALL
.STACK 100H

.DATA
    DT1      DB  1, 2, 'AB'
    COUNT    EQU $-DT1
    DT2      DW  1, 2, 'AB'
    DT3      DW  DT2
    DT4      DW  $-DT2

; 输入缓冲区
SN          DB  9, 0, 9 DUP(?)
NAM         DB  31, 0, 31 DUP(?)

; 提示信息（用 $ 结尾，供 09H 输出）
MSG1        DB  0DH, 0AH, 'Please input your student ID (8 digits): $'
MSG2        DB  0DH, 0AH, 'Please input your name pinyin: $'
MSG3        DB  0DH, 0AH, 'Your ID: $'
MSG4        DB  0DH, 0AH, 'Your Name: $'
CRLF        DB  0DH, 0AH, '$'

.CODE
MAIN PROC
    MOV AX, @DATA
    MOV DS, AX

;===== 显示 1 和 A =====
    MOV DL, DT1
    ADD DL, 30H                ; 1 → '1'
    MOV AH, 02H
    INT 21H

    MOV DL, DT1 + 2            ; 'A'
    MOV AH, 02H
    INT 21H
```

```

; 换行
LEA DX, CRLF
MOV AH, 09H
INT 21H

;===== 输入学号 =====
LEA DX, MSG1
MOV AH, 09H
INT 21H

LEA DX, SN
MOV AH, 0AH
INT 21H

;===== 输入姓名 =====
LEA DX, MSG2
MOV AH, 09H
INT 21H

LEA DX, NAM
MOV AH, 0AH
INT 21H

;===== 任务 6: 输出 =====
LEA DX, MSG3
MOV AH, 09H
INT 21H

; 输出学号
LEA SI, SN + 2
MOV CL, [SN + 1]
MOV CH, 0
OUTPUT_ID:
MOV DL, [SI]
MOV AH, 02H
INT 21H
INC SI
LOOP OUTPUT_ID

LEA DX, CRLF
MOV AH, 09H
INT 21H

```

```
LEA DX, MSG4
MOV AH, 09H
INT 21H
```

```
; 输出姓名
```

```
LEA SI, NAM + 2
MOV CL, [NAM + 1]
MOV CH, 0
ADD SI, CX
MOV BYTE PTR [SI], '$'
```

```
LEA DX, NAM + 2
MOV AH, 09H
INT 21H
```

```
LEA DX, CRLF
MOV AH, 09H
INT 21H
```

```
; 退出
```

```
MOV AX, 4C00H
INT 21H
```

```
MAIN ENDP
END MAIN
```