

【教材 6.1】 指出如下一段程序完成的功能：

```
IN    AL,20H
MOV   BL,AL
AND   AL,0FH
OUT   30H,AL
IN    AL,40H
MOV   CL,AL
AND   AL,0F0H
OUT   50H,AL
OR    BL,CL
MOV   AL,BL
MOV   DX,400H
OUT   DX,AL
```

【解答】

该程序段完成如下功能：

```
(20H) ∨ 0FH → 30H
(40H) ∨ 0F0H → 50H
(20H) ∨ (40H) → 400H
```

其中，20H,30H,40H,50H,400H 均为外设寄存器的地址。

【教材 6.2】 假定 PROG_n 为某一中断源的中断处理子程序入口地址，该中断源的类型号为 n,试编写程序段，将该入口地址填入中断向量表中相应位置。

【解答】

实现这一功能的程序段如下：

```
MOV AX,0
MOV ES,0 ; 将 ES 段寄存器赋初始值 0
```

CLI ;此时 CPU 关中断，禁止响应新的中断请求

MOV AX,SEG PROGN

MOV ES:[4*n+2],AX ; PROGN 的段地址 4*n+2 存储单元

MOV AX,OFFSET PROGN

MOV ES:[4*n],AX ; PROGN 的偏移地址 4*n 存储单元

STI ;设置好新的中断矢量表后，开中断

3. 改错 (共 10 个错误)

已知一 Win32 程序的功能是：在 BUF 存储区中有一个以 0 为结尾的字符串，将其中的小写字母转换成大写字母，然后用消息框显示出来。请修改该程序中的错误。

.386

.MODEL SMALL, STDCALL

OPTION CASEMAP:NONE

MessageBoxA PROTO :DWORD, :DWORD, :DWORD, :DWORD

ExitProcess PROTO :DWORD

includelib user32.lib

includelib kernel32.lib

.DATA

szDlgTitle DB 'A String After Change', 0

BUF DB 'mov ax, dx', 0aH, 0dH

DB 'ADD DX, digit', 0

COUNT = \$ - szDlgTitle

CHAR DB ?

MB_OK EQU 0

.CODE

BEGIN:

LEA ESI, BUF

NEXT: MOV ECX, COUNT

MOV CHAR, [ESI]


```

CMP    CHAR, 'a'

JLE    L1 ; 字符的比较应该用无符号数的条件转移指令 JB L1

CMP    CHAR, 'z'

JGE    L1 ; 字符的比较应该用无符号数的条件转移指令 JA L1

SUB     CHAR, 20H ; 应将修改后的数据放回 BUF 存储区，应为:

                        SUB BYTE PTR[ESI], 20H

L1:    INC     ESI
      LOOP    NEXT
      CALL    MessageBoxA, 0, ADDR BUF, ADDR szDlgTitle, MB_OK;

          ; 此种形式的调用应采用 INVOKE 命令，修改为：

          INVOKE MessageBoxA, 0, ADDR BUF, ADDR szDlgTitle, MB_OK;
      MOV     AH,     4CH

      INT     21H ; 这两条指令应修改为 WIN32 的 API 函数调用：

          INVOKE ExitProcess, 0

      END ; 应加上程序的起始地址：END BEGIN

```