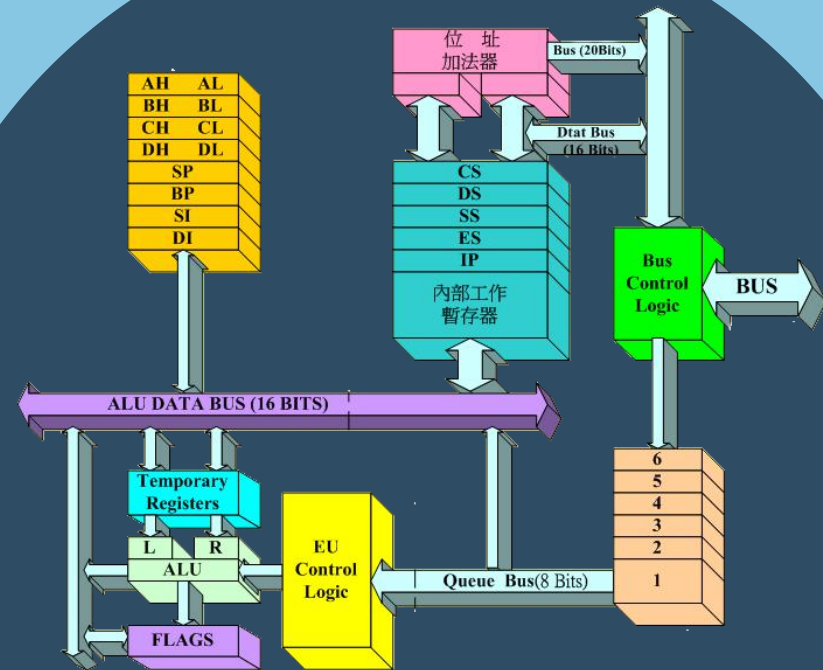


# 改写中断例程的方法

贺利坚 主讲



汇编语言程序设计  
Assembly Language

# 改写中断例程-以int 9为例

🖥️任务：安装一个新的int 9中断例程

🖥️功能：在DOS下，按F1键后改变当前屏幕的显示颜色，其他的键照常处理。

🖥️要解决的问题

## (1) 改变屏幕的显示颜色

改变从B800开始的4000个字节中的所有奇地址单元中的内容，当前屏幕的显示颜色即发生改变。

## (2) F1改变功能，其他键照常

可以调用原int 9中断处理程序，来处理其他的键盘输入

## (3) 原int 9中断例程入口地址的保存

要保存原int 9中断例程的入口地址原因：在新int 9中断例程中要调用原int 9中断例程  
保存在哪里？我们将地址保存在0:200单元处。

## (4) 新int 9中断例程的安装

我们可将新的int 9中断例程安装在0:204处。

```
mov ax,0b800h
mov es,ax
mov bx,1
mov cx,2000
s: inc byte ptr es:[bx]
add bx,2
loop s
```

# 实现方法

```
assume cs:code
stack segment
    db 128 dup (0)
stack ends
code segment
start:
```

; 设置各段地址

; 安装新程序

; 将原中断地址保存在0:200单元处

; 改变后中断的入口地址

```
mov ax,4c00h
int 21h
```

; 定义新中断例程

int9:

....

int9end:

```
code ends
end start
```

```
mov ax,stack ; 栈段
mov ss,ax
mov sp,128
push cs ; ds与cs相同
pop ds
mov ax,0 ; 附加段
mov es,ax
```

```
mov si,offset int9
mov di,204h
mov cx,offset int9end - offset int9
cld
rep movsb
```

```
call dword ptr cs:[200h]
```

此时, (CS)=0

```
push es:[9*4]
pop es:[200h]
push es:[9*4+2]
pop es:[202h]
```

```
cli
mov word ptr es:[9*4],204h
mov word ptr es:[9*4+2],0
sti
```

```
cmp al,3bh
jne int9ret
(F1的扫描码为3bh)
mov ax,0b800h
mov es,ax
mov bx,1
mov cx,2000
s: inc byte ptr es:[bx]
add bx,2
loop s
```

```
int9: push ax
      push bx
      push cx
      push es
```

```
in al,60h
```

```
pushf
```

; 调用旧中断例程

; 处理F1键

```
int9ret: pop es
        pop cx
        pop bx
        pop ax
        iret
```

```
int9end: nop
```