

- enter
 - pushl %ebp
 - movl %esp,%ebp
- leave
 - movl %ebp,%esp
 - popl %ebp

0x7c00处存放的代码

- 就是从磁盘引导扇区读入的那512个字节
- 引导扇区就是启动设备的第一个扇区
- 启动设备信息被设置在CMOS中...

CMOS: 互补金属氧化物半导体(64B-128B)。用来存储实时钟和硬件配置信息。

- 因此，硬盘的第一个扇区上存放着开机后执行的第一段我们可以控制的程序。

读入setup模块后: ok_load_setup

云课堂

```
Ok_load_setup: //载入setup模块
    mov dl,#0x00    mov ax,#0x0800 //ah=8获得磁盘参数
    int 0x13        mov ch,#0x00    mov sectors,cx
    mov ah,#0x03    xor bh,bh        int 0x10 //读光标
    mov cx,#24      mov bx,#0x0007    7是显示属性!
    mov bp,#msg1    mov ax,#1301     int 0x10 //显示字符
    mov ax,#SYSSEG  //SYSSEG=0x1000
    mov es,ax
    call read_it    //读入system模块
    jmp 0,SETUPSEG
```

显示这24个字符将是大家的第一个“创举”!

转入0x9020:0x0000
执行setup.s

- boot工作:读setup,
读system...

bootsect.s中的数据 //在文件末尾

```
sectors: .word 0 //磁道扇区数
msg1:.byte 13,10
.ascii "Loading system..."
.byte 13,10,13,10
```



// 调用者

...

call target

...

call指令:

- 1) 将eip中下一条指令的地址A保存在栈顶
- 2) 设置eip指向被调用程序代码开始处

//建立被调用者函数的堆栈框架

pushl %ebp

云课堂

movl %esp, %ebp

//被调用者函数体

//do sth.

...

//拆除被调用者函数的堆栈框架

movl %ebp, %esp

popl %ebp

ret

将地址A恢复到eip中

SAMSUNG

R429