对于下⾯两⾏代码：

**mov ax, BootMessage**

**mov bp, ax**

思考为什么 mov bp, ax 后，int 10h 就能够取到 BootMessage 了？运⾏到这⾏代码的时候 *ax* ⾥⾯的值是多少？这个值是不是 BootMessage 所在内存中的位置（即相对地址还是绝对地址）？

10H 中断用于显示字符串，bp 指向要显示的字符串，调用中断就显示了字符串。

ax中的值是 BootMessage 对应的字符串的首地址。这个值是 BootMessage 的绝对地址。

1. boot.asm ⽂件中，**org 0700h** 的作⽤

告诉汇编器，当前这段代码会放在07c00h处。所以，如果之后遇到需要绝对寻址地指令，那么绝对地址就是07c00h加上相对地址。在第一行加上org 07c00h只是让编译器从相对地址07c00h处开始编译第一条指令，相对地址被编译加载后就正好和绝对地址吻合

1. 为什么要把 boot.bin 放在第⼀个扇区？直接复制为什么不⾏？

BIOS程序检查软盘0面0磁道1扇区，如果扇区以0xaa55结束，则认定为引导扇区，将其512字节的数据加载到内存的07c00处，然后设置PC，跳到内存07c00处开始执行代码。以上的0xaa55以及07c00都是一种约定，BIOS程序就是这样做的，所以我们就需要把我们的os放在软盘的第一个扇区，填充，并在最末尾写入0xaa55。

直接拷贝是不可以的（普通的读写操作（mv, rm,cp）是基于文件系统的，文件系统是一个逻辑概念。引导扇区，是磁盘第一个磁道的第一个扇区，他是一个物理概念，在文件系统中，这个扇区是不可见的。 ）

1. loader 的作⽤有哪些?

跳入保护模式

启动内存分页

1. L1，L6 各标识了⼀个字节（8bit）的数据，eax 是⼀个 16 位寄存器，说明下⾯每⾏代码的意思。

⾏号 代码

* 1. mov al, [L1] 将L1中数据送至寄存器
  2. mov eax, L1 将L1首地址送至寄存器
  3. mov [l1], ah 将L1的值变为0
  4. mov eax, [L6] 将L6的值送至寄存器
  5. add eax, [L6] 加上L6
  6. add [L6], eax 将eax地址放到L6
  7. mov al, [L6] 将L6的值送至寄存器

1. **times 510-($-$$) db 0**

为什么是 510? $ 和 $$ 分别表⽰什么？不⽤ times 指令怎么写（等价命令）？

0XAA55占两个字节

$ 表示当前行被汇编后的地址，$ $ 表示一个节的开始处被汇编后的地址。这句话表示本行距离程序开始处的相对距离。

Mov eax, $

Mov ecx, **510-($-$$)**

**Mov eax, 0**

**Sub ecx, 1**

**Cmp ecx, 0**

**jne**

1. 解释 db 命令：**L10 db** “**w**”**,** “**o**”**,** “**r**”**,** “**d**”**, 0** 这条语句的意

义，并且说明数字 0 的作⽤。

用4个bytes存放word，0是字符串的最后一位。

1. **L1 db 0**

**L2 dw 1000**

L1、L2 是连续存储的吗？即是否 L2 就存储在 L1 之后？

是连续存储的

1. 要是不知道 L6 标识的是多⼤的数据，下⾯这句话对不对？ **mov [L6], 1**

不对

1. 如何处理输⼊输出？在代码中哪⾥体现出来？

调用中断

1. 通过什么来保存前⼀次的运算结果？在代码中哪⾥体现出来？

申请的具体内存

1. 随机选择代码段，说明作⽤。
2. 有哪些段寄存器？

8086/8088将1MB(2^20)地址空间分成若干逻辑段  当前使用的段值存放在段寄存器中，由段值和段内位移形成20位地址

代码段寄存器CS(Code Segment)

数据段寄存器DS(Data Segment)

堆栈段寄存器SS(Stack Segment)

附加段寄存器ES(Extra Segment)

1. 8086/8088 存储单元的物理地址长，CPU 总线的数量，可以直接寻址的物理地址空间。

存储单元的物理地址长20位，范围是00000H至FFFFFH

8086/8088CPU有20根地址总线，可以直接寻址的物理地址空间是1M字节（=220）

ALU最多进行16位运算，存储单元地址偏移的指针寄存器都是16位

存储器分段和使用段寄存器寻址1M字节物理空间



1. 如何根据逻辑地址计算物理地址？

根据逻辑地址计算物理地址  物理地址=段值\*16+偏移（左移四位）

1. 寄存器的寻址⽅式（知道如何计算）。

立即寻址⽅方式 MOV AX, 1234H

寄存器寻址⽅方式 MOV SI,AX

直接寻址⽅方式 MOV AX,[1234H] ;设寄存器DS的内容是5000H

寄存器间接寻址⽅方式 MOV AX,[SI] ;(DS)=5000H,(SI)=1234H

寄存器相对寻址⽅方式 ;(DS)=5000H,(DI)=3678H MOV AX,[DI+1223H]

基址加变址寻址⽅方式 ;(DS)=5000H,(BX)=1223H,(DI)=54H MOV AX,[BX+DI]

相对基址加变址寻址⽅方式

MOV AX,[BX+DI-2] ;(DS)=5000H,(BX)=1223H,(DI)=54H

1. ⼏个常⽤指令的作⽤（如 MOV，LEA 等）。
2. 主程序与⼦程序的⼏种参数传递⽅式。

利用寄存器传递参数

利用约定存储单元传递参数

利用堆栈传递参数

利用CALL后续区传递参数