1. boot.asm ⽂件中，**org 0700h** 的作⽤

org是伪指令，不生成对应的二进制指令，只是汇编器使用的；此指令告诉汇编器当前的代码会放在07c00h处，如果需要绝对地址，则为07c00h+相对地址。

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程序在运行时会先检查软盘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512B数据加载到内存。

普通的读写操作事基于文件系统的，文件系统是一个逻辑概念。而引导扇区是一个物理概念，在文件系统中不可见。

1. loader 的作⽤有哪些?

跳入保护模式；启动内存分页，将内核kernel加载入内存，突破了512B的限制。

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。
   1. mov al, [L1] ； 将L1地址中的值放入al中
   2. mov eax, L1 ； 将L1地址放入eax中
   3. mov [l1], ah ； 将ah中的内容赋值给l1
   4. mov eax, [L6] ； 将L6地址中的值放入eax中
   5. add eax, [L6] ； eax中的值=eax中的值+L6中的值
   6. add [L6], eax ； L6中的值 = L6中的值+eax中的值
   7. mov al, [L6] ； 将L6地址中的值放入al中
2. **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）？

填充剩下的空间，使生成的二进制代码恰好为512B；

$表示当前地址

$$表示段开始地址

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。

等价于 L10： db “word”

0代表字符串的结束

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？

不是，数据段空间大小为定义数据所需的16B的最小整数倍？

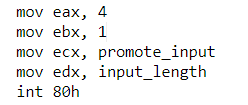
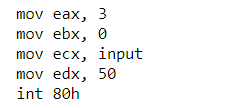
1. 要是不知道 L6 标识的是多⼤的数据，下⾯这句话对不对？ **mov [L6], 1**

对

1. 如何处理输⼊输出？在代码中哪⾥体现出来？

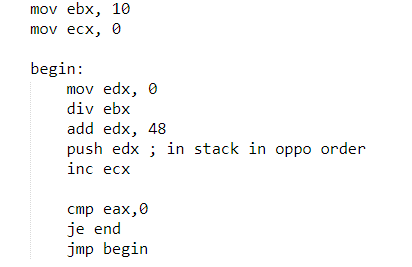
……

系统调用号

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？

edx？



1. 随机选择代码段，说明作⽤。

……

1. 有哪些段寄存器？

CS代码段register；DS数据段register；SS堆栈段register；ES附加段register

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。

20bit；20根地址线；

8/16根数据线

可直接寻址的物理地址空间=2­20­=1MB

1. 如何根据逻辑地址计算物理地址？

物理地址由两部分组成：段基址(段起始地址高16位)和[偏移地址](https://www.baidu.com/s?wd=%E5%81%8F%E7%A7%BB%E5%9C%B0%E5%9D%80&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1d-P1DkmvmLn1csnvDkP1w-0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnW6kPjfLrjb4)。前者由段寄存器给出，后者是指存储单元所在的位置离段起始地址的偏移距离。当CPU寻址某个存储单元时，先将段寄存器的内容左移4位，然后加上指令中提供的16位[偏移地址](https://www.baidu.com/s?wd=%E5%81%8F%E7%A7%BB%E5%9C%B0%E5%9D%80&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1d-P1DkmvmLn1csnvDkP1w-0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnW6kPjfLrjb4)而形成20位物理地址。在取指令时，CPU自动选择代码段寄存器CS，左移4位后，加上指令提供的16位[偏移地址](https://www.baidu.com/s?wd=%E5%81%8F%E7%A7%BB%E5%9C%B0%E5%9D%80&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1d-P1DkmvmLn1csnvDkP1w-0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnW6kPjfLrjb4)，计算出要取指令的物理地址。堆栈操作时，CPU自动选择堆栈段寄存器SS，将其内容左移4位后，加上指令提供的16位偏移地址，计算出栈顶单元的物理地址。每当存取操作数时，CPU会自动选择数据段寄存器(或附加段寄存器ES)，将段基值左移4位后加上16位偏移地址，得到操作数在内存的物理地址。

1. 寄存器的寻址⽅式（知道如何计算）。

……

1. ⼏个常⽤指令的作⽤（如 MOV，LEA 等）。

……

1. 主程序与⼦程序的⼏种参数传递⽅式。

常见的的参数传递有4种

1.寄存器法

寄存器法就是将入口参数和出口参数存放在约定的寄存器中。

优点：数据传递书读快、编程较方便、节省内存单元。

缺点：当传递参数过多时候，由于寄存器个数有限，及寄存器的频繁使用，将导致寄存器不足。

适用：参数较少的子程序

2.约定单元法：

约定单元法顾名思义是吧入口参数和出口参数都放在事先约定好的单元中

优点：不占用寄存器、参数个数任意、每个子程序要处理的数据和送出的结构都有独立的存储单元

缺点：但用一定数量的存储单元，增加编程中对变量定义的难度

3.堆栈法

堆栈法是利用堆栈来传递参数的。

优点：参数不占用寄存器，和存储单元。参数存放在公共堆栈区，处理完后客恢复。参数个数一般不限

缺点：由于参数和子程序混杂在一起，存取参数时候必须小心计算它在堆栈中的位置。要注意断点的保存和恢复。

4.地址表法

这种方法是把参数组成的一张参数表放在某个存储区中，然后只要主程序和子程序约定好这个存储区的首地址和存放的内容，在主程序中将参数传递给地址表，在子程序中根据地址表给定的参数就可以完成操作。