操作系统实验（一）问答题参考

南京大学软件学院

2015.3

实验重点

本次作业重点在于熟悉掌握：8086 寻址方式和指令系统，主程序和子程序的参数传递以及 $nasm + bochs$ 实验平台的搭建和使用

1 问题清单

在整个实验的过程中，无论是编程还是查资料，请各位同学注意思考以下问题，助教检查时会从中随机抽取数个题目进行提问，根据现场作答给出分数。请注意，我们鼓励自己思考和动手实验，如果能够提供自己的思考结果并辅助以相应的实验结果进行说明，在分数评定上会酌情考虑。

1. boot.asm 文件中，**org 0700h** 的作用

   参考答案：

   告诉汇编器该段代码会被加载到内存的 07c00 处，当编译的时候遇到相对寻址的指令的时候会用 07c00 加上相对地址得到绝对地址，

2. 为什么要把 boot.bin 放在第一个扇区？直接复制为什么不行?

   参考答案：

   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。

   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。而引导扇区，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。

3. loader 的作用有哪些?

   参考答案：

   加载内核入内存，跳入保护模式，内存分页。

4. 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]
2	mov eax, L1
3	mov [l1], ah
4	mov eax, [L6]
5	add eax, [L6]
6	add [L6], eax
7	mov al, [L6]

参考答案：

1 mov al, [L1] ; copy byte at L1 into AL

2 mov eax, L1 ; EAX = address of byte at L1

3 mov [L1], ah ; copy AH into byte at L1

4 mov eax, [L6] ; copy double word at L6 into EAX

5 add eax, [L6] ; EAX = EAX + double word at L6

6 add [L6], eax ; double word at L6 += EAX

7 mov al, [L6] ; copy first byte of double word at L6 into AL

5. **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)?

参考答案：

因为需要填充 512 个字节的数据，最后两个字节是以 0xaa55 结尾，所以需要填充 510 个字节 $ 表示当前的字节数，$$ 表示开始的字节。

不用 times 命令可以使用 db 0 循环 ($-$$) 次

6. 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。

参考答案：

填充字符串“word”，最后的 0 表示结束符，即在 C/C++ 里字符串末尾的’
0’ 字符。

7. **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？

   参考答案：L2 就存储在 L1 之后

8. 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

   参考答案：

   This statement produces an operation size not specified error. Why? Because the assembler does not know whether to store the 1 as a byte, word or double word. To fix this, add a size specifier: mov dword [L6], 1 ; store a 1 at L6 This tells the assembler to store an 1 at the double word that starts at L6. Other size specifiers are: BYTE, WORD, QWORD and TWORD.

9. 如何处理输入输出？在代码中哪里体现出来？

   参考答案：

   使用中断处理。

10. 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？

    参考答案：

    栈或者寄存器

11. 随机选择代码段，说明作用。

12. 有哪些段寄存器？

    参考答案：

    代码段，数据段，堆栈段，附加段

13. 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。

    参考答案：

    20、20、1M

14. 如何根据逻辑地址计算物理地址？

    参考答案：

    物理地址 = 段值 *16+ 偏移（左移四位）（保护模式下如此，其他模式下 16 回变化）

15. 寄存器的寻址方式（知道如何计算）。

    参考答案：

    立即寻址方式；寄存器寻址方式；直接寻址方式；寄存器间接寻址方式；寄存器相对寻址方式；基址加变址寻址方式；相对基址加变址寻址方式

16. 几个常用指令的作用（如 MOV，LEA 等）。

    参考答案：

    MOV: 把一个字或字节从源操作数 SRC 送至目的操作数 DST

    LEA: 把操作数 OPRD 的有效地址传送到操作数 REG

    PUSH、POP: 堆栈操作指令

    ADD、ADC、SUB、SBB: 加减运算指令

    ——其他见 PPT, 更详细的请翻阅《80X86》

17. 主程序与子程序的几种参数传递方式。

    参考答案：

    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

2 参考资料

1. 《Orange'S: 一个操作系统的实现》
2. NASM doc
3. Introduction to NASM
4. MASM Tutorial