

操作系统实验（一）

南京大学软件学院

2015.3

实验重点

本次作业重点在于熟练掌握：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 实验内容

1.1 Hello OS

选择任意你喜欢的平台（可以是 mac、windows 或其他），参考 PPT，搭建 *nasm + bochs* 实验平台，在该实验平台上汇编 boot.asm，并用 bochs 执行，显示 Hello OS，请提交运行截图和代码。

1.2 汇编语言实践

参考寻址方式和指令系统 PPT，熟悉汇编指令，用汇编语言实现斐波那契数列，具体要求如下：

- 系统请求输入一个正整数，用户输入需要显示的项的个数，回车键结束输入。
- 显示指定数目的斐波那契数列项，要求各个项用空格隔开，并用不同颜色显示（不限制颜色的种类、数目）。
- 计算达到 25 项可能会超出 16 位二进制数，如果实现了超过 25 项的计算可以加分。
- 鼓励使用高级指令（可加分）。
- 如果有其他亮点，无论是功能丰富还是代码实现上，检查时请向助教提出，酌情进行加分。
- 请提交运行截图和代码。
- 一个可能的运行结果如下图所示：

```
*****
OS_Lab_1 sample Fibonacci Sequence
please input n = 6
Fibonacci Sequence is: 1 1 2 3 5 8
*****
```

1.3 代码阅读

仔细阅读《Orange's》的第一章和第二章，深入理解 boot.asm 文件中的代码，尤其注意问题清单中的问题。

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）？

- 这道题目通过阅读 Orange's 或者其他资料即可得到答案，检查时只要说出正确答案就可以通过。如果你只是完成了这项要求，请提交一个 pdf 文档，文档中字数少于 50 字。
- 鼓励同学们进行实验验证，请思考自己认为正确地结论，并通过实验手段进行验证，如果你的验证是可行的、有效的、或者至少是能体现思考的，你将获得一定的**加分**。完成这项要求的同学，提交上一项的文档以及实验截图。助教检查时，请主动出示截图，并讲解你的实验思路和结果，随机选择同学现场演示其实验过程。

2 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
2. 为什么要把 boot.bin 放在第一个扇区？直接复制为什么不行？
3. loader 的作用有哪些？
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]

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)?

6. 解释 db 命令: **L10 db “w”, “o”, “r”, “d”, 0 ;0** 这条语句的意义, 并且说明数字 0 的作用。

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?

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

9. 如何处理输入输出? 在代码中哪里体现出来?

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?

11. 随机选择代码段, 说明作用。

12. 有哪些段寄存器?

13. 8080/8086 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。

14. 如何根据逻辑地址计算物理地址?

15. 寄存器的寻址方式 (知道如何计算)。

16. 几个常用指令的作用 (如 MOV, LEA 等)。

17. 主程序与子程序的几种参数传递方式。

3 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](#)