

2018 操作系统实验（一）

本次实验重点在于熟悉掌握：8086 寻址方式和指令系统，x86 汇编基础，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 Windows, Mac 或 Linux），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。

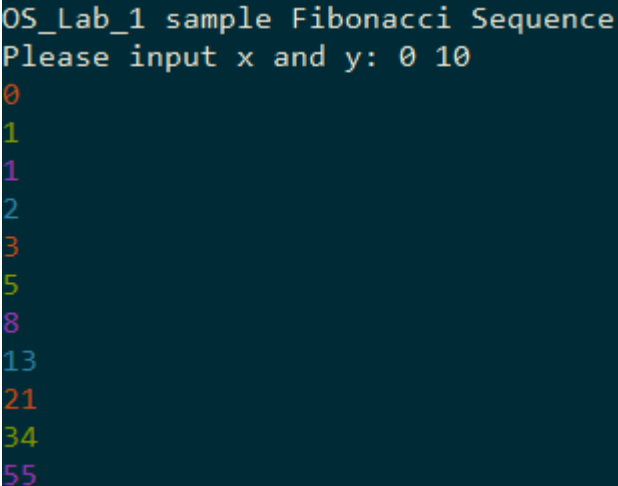
1.2 汇编语言实践

参考寻址方式和指令系统 PPT，熟悉汇编指令，用汇编语言（NASM）实现斐波那契数列。

输入输出格式

- 输入为两个整数 x, y ($0 \leq x \leq y$)，以空格分割，回车结束输入。
- 输出为斐波那契数列中的第 x 至 y 项（包括 x, y ），每行一个数字，以不同颜色显示。
- 程序使用标准输入（键盘）和标准输出（屏幕）。

样例



```
OS_Lab_1 sample Fibonacci Sequence
Please input x and y: 0 10
0
1
1
2
3
5
8
13
21
34
55
```

评分标准

- 基本得分：实现上述的基本功能
- 附加得分：实现超过 64 位的整数运算，即支持大于第 200 项的输出

注意事项

- $f(0) = 0, f(1) = 1$.
- 本实验要求在 Linux/Windows/macOS 系统上面完成，而不是在 bochs 内。

- 请提交运行截图和代码。

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。如果你只是完成了这项要求，请将回答写入一个 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。完成这项要求的同学，请将实验过程及截图附加在上一项的文档中。

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。

行号	代码
1	<code>mov al, [L1]</code>
2	<code>mov eax, L1</code>
3	<code>mov [L1], ah</code>
4	<code>mov eax, [L6]</code>
5	<code>add eax, [L6]</code>
6	<code>add [L6], eax</code>
7	<code>mov al, [L6]</code>

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，说明上面每行代码的意思。
5. `times 510-($-$$) db 0` 中的数字为什么是 510？`$` 和 `$$` 分别表示什么？不用 `times` 指令怎么写（等价命令）？
6. 解释 `L10 db "w", "o", "r", "d", 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` 中的 `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. 有哪些段寄存器？
12. 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。
13. 如何根据逻辑地址计算物理地址？
14. 寄存器的寻址方式（知道如何计算）。
15. 几个常用指令的作用（如 MOV，LEA 等）。
16. 主程序与子程序的几种参数传递方式。

3. 参考资料

- 《Orange's: 一个操作系统的实现》
- NASM Tutorial - <https://cee.github.io/NASM-Tutorial/>
- NASM Docs - <http://www.nasm.us/doc/>

如遇到实验相关问题，请在 Moodle 上发帖或向助教发邮件。