操作系统实验（一）

南京大学软件学院

2017.3

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

检查时间： 约2-3周后

# 实验内容

## 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。

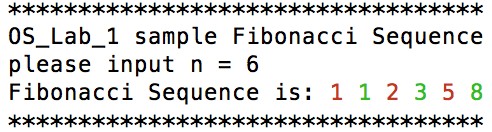
**1.2** 汇编语言实践

参考寻址⽅式和指令系统 PPT，熟悉汇编指令，⽤汇编语⾔（NASM）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。
* 注意本次实验要求在 linux（建议安装32位系统，当然学长亲测64位系也是可以的）/mac/win 系统上⾯完成，⽽不是在 bochs 内。
* ⼀个可能的运⾏结果如下图所⽰：



* 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，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。

# 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]

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）？

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。

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？

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**
2. 如何处理输⼊输出？在代码中哪⾥体现出来？
3. 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？
4. 随机选择代码段，说明作⽤。
5. 有哪些段寄存器？
6. 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。
7. 如何根据逻辑地址计算物理地址？
8. 寄存器的寻址⽅式（知道如何计算）。
9. ⼏个常⽤指令的作⽤（如 MOV，LEA 等）。
10. 主程序与⼦程序的⼏种参数传递⽅式。

# 建议：

实验可参照以下资料学习汇编语言：

1.[NASM doc](http://www.nasm.us/doc/) 官方文档[PDF版本](https://github.com/czqInNanjing/SmallProject/blob/master/books/nasmdoc.pdf)

以下两份资料可用于入门

2. [Introduction to NASM](https://github.com/czqInNanjing/SmallProject/blob/master/books/nasm.pdf)

3. NASM Tutorial (http://cs.lmu.edu/~ray/notes/nasmtutorial/)

# Q & A

1.问：可否假定程序输入的参数是递增的？即，只可能出现“1 3 5 10 42”，不会出现“10 42 3 5 1”？    
  
  
答：可以假定输入参数是递增的，不用考虑“10 42 3 5 1”这样的乱序输入。 一次输入所包含的参数数量不会超过10个。   
  
  
2.问：使用虚拟机和在真机上安装Linux都可以吗？    
  
  
答：当然可以。装系统是过程，不是目的。本次作业的强制要求只是使用NASM汇编语言。如果你有能力的话，完全可以在宿舍空调的遥控器上编程完成每一次作业。（假设该遥控器上可以使用NASM汇编语言的话）    
  
  
3.问：安装双系统有什么风险？    
  
  
答：包括但不限于：硬盘数据丢失，原系统无法引导等。大部分意外是可逆的，通过你孜孜不倦的探索是可以弥补的。在探索的过程中正好可以体会一下为什么有人说“开源的是最贵的”。    
  
  
4.问：安装64位Ubuntu不行吗？    
  
  
答：还是这句话，装系统是过程不是目的。我们建议不要使用64位Ubuntu是因为往届学长说64位Ubuntu在进行什么操作的时候会出现什么问题，学长也可能是听学长的学长说的，所以说只是一点长者的经验。如果你选择面对这种风险，我们肯定会资瓷。如果你能证明根本没有这种风险，或者找到这点小问题的解决方法，当然是坠吼的。    
  
  
5. 问：不能用c语言吗？    
  
答：不能。这次不能,包括调用C类库如Printf之类的。后面几次实验会让你尽情使用c语言的，保证量足够大，让你用得足够开心。另外为了防止以后重复回答同类问题，这里一并回答：不光c不能用，连java、perl、python、c#、lua、c++也不能用。    
  
6. 问：64位系统中遇到xxxx问题怎么办？    
  
答：请参考以下链接：

http://cs.lmu.edu/%7Eray/notes/nasmtutorial/   
中文版http://cee.github.io/NASM-Tutorial/（用于参考，没有更新到最新版本）

7. 问：怎么用NASM输出文字？    
  
答：请百度”NASM helloworld”，我们不太资瓷大家问一些能在百度第一页第一条找到答案的问题。另外，谷歌比百度更好用，手册（即文档，学会查看官方文档比频繁查google不知道高到哪里去）比谷歌更好用。

如有其它问题请在CMS上发帖或向助教发邮件！