# 2023年春操作系统实验（一）

**预计检查日期：2023年4月6日**

## 实验任务

本次实验共有3个小任务，满分7分，不设置附加分项目。你需要讲本次实验的所有产物放在一个名为`学号_姓名_lab1`的目录下，每个任务单独设置一个子目录。在实验检查前你需要把该目录打包上传到Moodle上。

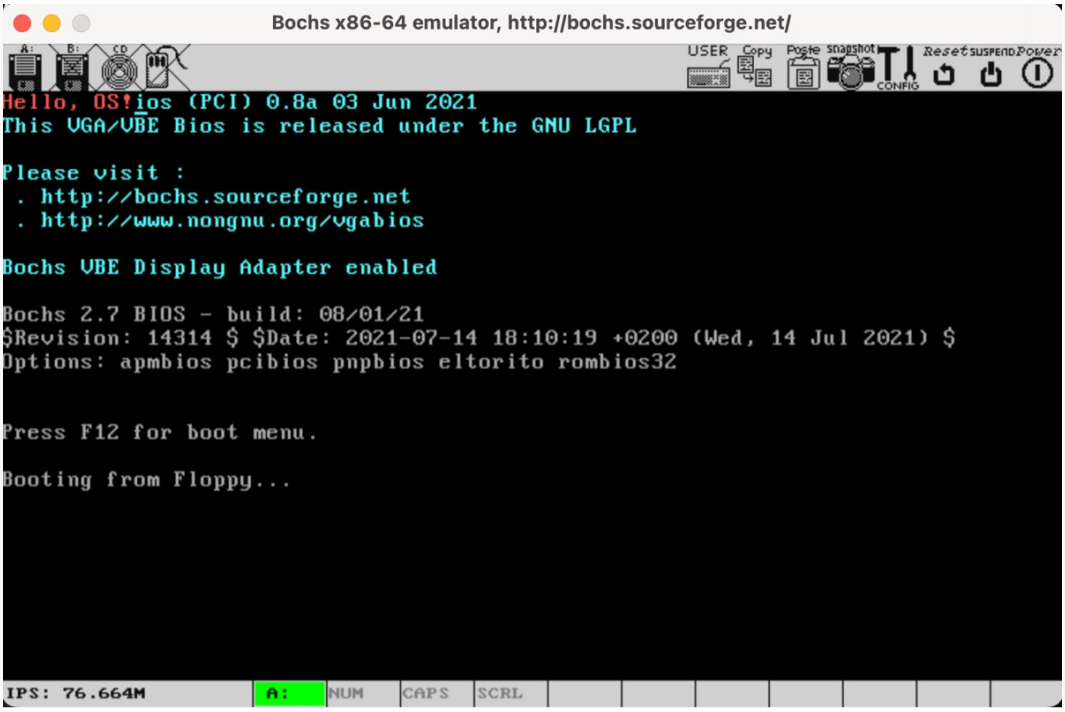
示例：

```
.
├── t1
│   ├── a.img
│   ├── bochsrc
│   ├── boot.asm
│   ├── boot.bin
│   └── screenshot_t1.png
├── t2
│   ├── a.img
│   ├── bochsrc
│   ├── boot.asm
│   ├── boot.bin
│   ├── loader.asm
│   ├── loader.bin
│   └── screenshot_t2.png
└── t3
    ├── main.asm
    ├── main.bin
    └── screenshot_t3.png

4 directories, 15 files
```

## 实验任务1: Hello OS（1分）

选择任意平台，参考讲义搭建NASM+Bochs实验平台，在该实验平台上汇编`boot.asm`并用Bochs虚拟机运行，显示“Hello OS!”。最终运行结果如下图所示：




### 具体要求

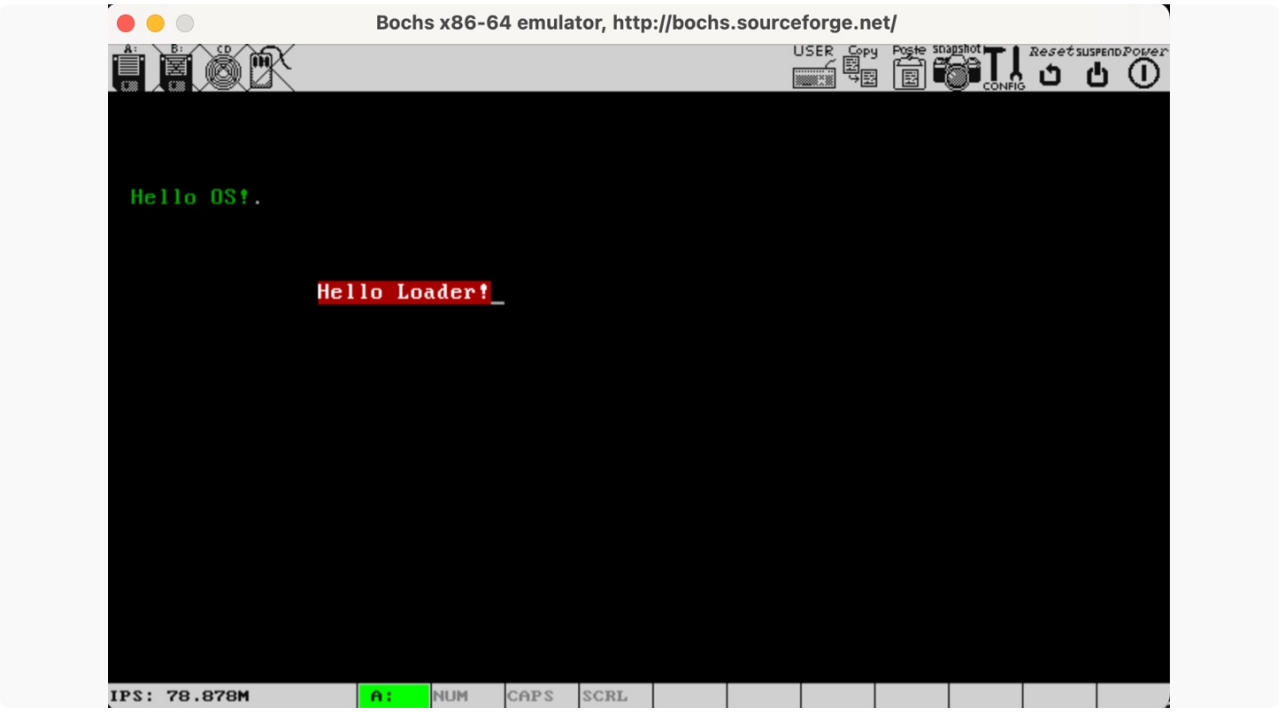
- 理解`boot.asm`、`bochsrc`中的内容；理解实验过程中涉及的所有命令、选项、参数的含义。
- **提交内容**：源代码、二进制文件、制作的镜像文件、运行截图

### 评分标准

- 在不依赖助教提示、查看讲义的情况下复现实验，能解释`boot.asm`中的代码，得1分；
- 部分操作需要依赖助教提示或查看讲义，或是部分代码无法解释，得0分。

## 实验任务 2: 从 Boot 到 Loader（2 分）

阅读《一个 64 位操作系统的设计与实现》3.1 章（或《Orange's 一个操作系统的实现》4.1 章）的内容，书中的 Boot 程序把控制权交给 Loader 程序。请理解这一部分代码，并把这些代码加入到你的系统中。其中 Boot 程序需要在屏幕上显示 `Hello OS!` 的字样，要求先空 4 行 2 列，颜色为绿色；Loader 程序显示 `Hello Loader!` 的字样，要求先空 8 行 16 列，颜色为白色，背景为红色。在输出前需要清屏，不能出现类似实验任务 1 中的无关字样。最终运行结果如下图所示：




### 具体要求

- 理解 `boot.asm` 、 `bochsrc` 、 `loader.asm` 中的内容；理解实验过程中涉及的所有命令、选项、参数的含义；
- 这一部分代码较多，不要求原创，但写在你作业中的代码一定要理解；有关 FAT12 的内容是下次实验的内容，本次实验大致了解即可。
- **提交内容**：源代码、二进制文件、制作的镜像文件、运行截图

### 评分标准

- 在不依赖助教提示、查看讲义的情况下复现实验，能解释相关代码的内容，且输入输出符合题目要求，得 2 分；
- 输出格式不符合要求，无法理解部分代码的内容，视情况扣 1-2 分。

## 实验任务 3: 整数除法（4 分）

使用 NASM 汇编语言实现整数除法。

输入：被除数$x$和除数$y$，其中$0 \leq x, y \leq 10^{100}$

输出：计算$x \div y$的结果，包括商和余数。若出错则输出错误信息。

示例：

输入：`10 20`

输出：`0 10`

### 具体要求

- **不得抄袭**；
- 本实验在本机的 Linux/macOS/Windows 等系统上完成，而不是在 Bochs 中完成；
- 只能使用 NASM 实现，不可使用其他语言；
- 必要之处给出注释。
- **提交内容**：源代码、二进制文件、运行截图、说明文档（PDF 格式，非必要）

### 评分标准

- 完成所有要求得 4 分；
- 代码质量不高视情况扣 1-4 分；
- 无法解释代码视为抄袭，本次实验得 0 分。

## 实验问题

在整个实验过程中，无论是编程还是查资料，请同学们注意思考以下问题，助教检查时会从中随机抽取若干题目进行提问，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。实验问题满分3分。

1. 8086有哪5类寄存器？请分别举例说明其作用。
2. 有哪些段寄存器，它们的作用是什么？
3. 什么是寻址？8086有哪些寻址方式？
4. 主程序与子程序之间如何传递参数？
5. 解释 `boot.asm` 文件中 `org 07c00h` 的作用。如果去掉这一句，整个程序应该怎么修改？
6. 解释 `int 10h` 的功能。
7. 解释 `boot.asm` 文件中 `times 510-($-$$) db 0` 的作用。
8. 解释 `bochsrc` 中各参数的含义。
9. `boot.bin` 应该放在软盘的哪一个扇区？为什么？
10. 为什么不让Boot程序直接加载内核，而需要先加载Loader再加载内核？
11. Loader的作用有哪些？
12. Kernel的作用有哪些？

## 参考资料

- 《一个64位操作系统的设计与实现》
- 《Orange's一个操作系统的实现》
- NASM Docs https://www.nasm.us/doc/

如遇到实验相关问题，请在课程群中咨询。