

# 操作系统实验报告

---

计科（大数据、人工智能方向）17341155 王永康

## （实验二：具有加载COM格式用户程序的监控程序的设计）

### 一、实验目的

实现一个简单的批处理的监控程序，同时熟悉BIOS的调用。

### 二、实验内容

- 设计简单监控程序，可以从引导盘加载其他用户程序。
- 设计四个（或更多）有输出的用户可执行程序

设计四个有输出的用户可执行程序，分别在屏幕1/4区域动态输出字符，如将用字符'A'从屏幕左边某行位置45度角下斜射出，保持一个可观察的适当速度直线运动，碰到屏幕相应1/4区域的边后产生反射，改变方向运动，如此类推，不断运动；在此基础上，增加你的个性扩展，如同时控制两个运动的轨迹，或炫酷动态变色，个性画面，如此等等，自由不限。还要在屏幕某个区域特别的方式显示个人的学号姓名等个人信息。

### 三、实验方案

#### 操作系统实验工具与环境

- 实验支撑环境
  - 硬件：个人计算机
  - 主机操作系统：Linux(Ubuntu 18.04)
  - 虚拟机软件：Bochs
- 实验开发工具：
  - 汇编语言工具：x86汇编语言
  - 汇编编译工具：NASM+GCC
  - 调试工具：Bochs

#### 方案的思想

- **监控程序的实现：**通过调用BIOS的中断功能，显示必要提示信息后，读取用户的键盘输入，以确定从引导盘的特定扇区加载一个开发好的COM格式的可执行程序到指定的内存位置，然后启动这个程序，实现操作系统执行用户程序这一项功能。
- **四个可执行程序的实现：**
  - 飞翔字符在左下1/4区弹跳的程序。
  - 显示名字、学号信息的程序在右下1/4区的程序。
  - 弹跳的酷炫字符在右1/4区的程序。
  - 清屏程序：可以清空界面到操作系统初始界面的程序。

## 相关原理

- 监控程序的原理：

由实验一设计的引导程序的原理，把监控程序作成引导程序，实现计算机的控制权转移到监控程序这一步；之后，通过调用BIOS的int 1H中断实现对用户的键盘输入的读取，根据返回的字符存在寄存器//中，//网址

将ASCII码转换成整数，存入寄存器AL中，决定读取的扇区号，再利用BIOS的int 13H中断的功能（具体功能见下图），读取特定的扇区，把程序加载至内存中；

## BIOS 的13H调用

- BIOS 的13H提供了磁盘读写的调用
- BIOS的13H号调用功能与参数

读扇区	13H	02H	AL: 扇区数(1~255) DL: 驱动器号(0 和 1 表示软盘, 80H 和 81H 等表示硬盘或 U 盘) DH: 磁头号(0~15) CH: 柱面号的低 8 位 CL: 0~5 位为起始扇区号(1~63), 6~7 位为硬盘柱面号的高 2 位(总共 10 位柱面号, 取值 0~1023) ES:BX: 读入数据在内存中的存储地址	返回值: ■ 操作完成后 ES:BX 指向数据区域的起始地址 ■ 出错时置进位标志 CF=1, 错误代码存放在寄存器 AH 中 ■ 成功时 CF=0、AL=0
-----	-----	-----	---	--

 中山大学 计算机科学系 操作系统课程组 凌应标制作 ©2015年3月

再利用jmp 0A100h指令，这样就实现了把控制权交给用户程序。

- 用户程序的设计原理：

首先，汇编中利用org 0A100h指令把文件加载到特定内存区域，最后运行完程序，再利用指令jmp 07c00h，把控制权交还给监控程序。

具体程序的设计原理，主要调用了BIOS的相关显示中断功能，详情见代码。

## 四、实验过程和结果

## 监控程序：

- 编写监控程序源文件myOS.asm
- 利用nasm进行编译，终端下输入

```
1 nasm myOS.asm -o myOS.bin
```

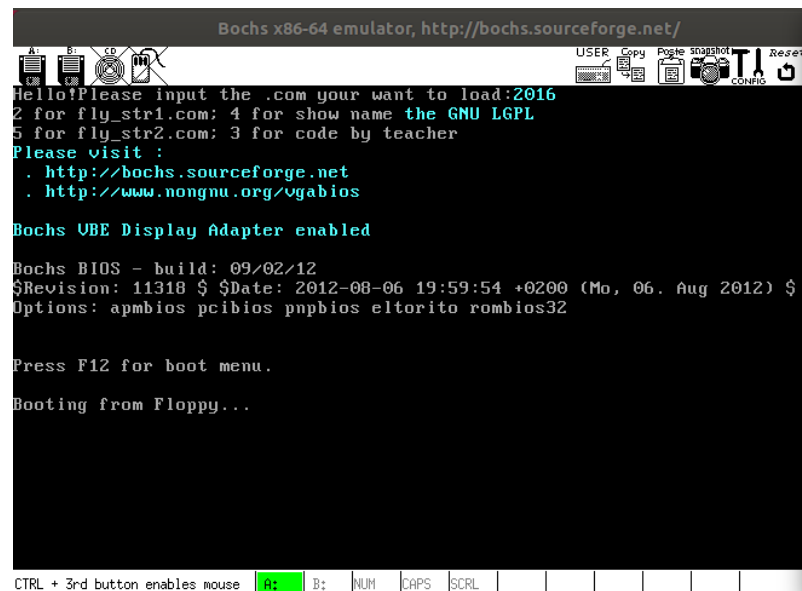
生成myOS.bin文件。

- 然后再终端利用Linux的dd命令，

```
1 dd if=myOS.bin of=a.img seek=0 bs=512 count=1  
conv = notrunc
```

把myOS.bin文件写入a.img软盘的首扇区；

- 之后运行Bochs，终端输入Bochs，输入c,运行虚拟机



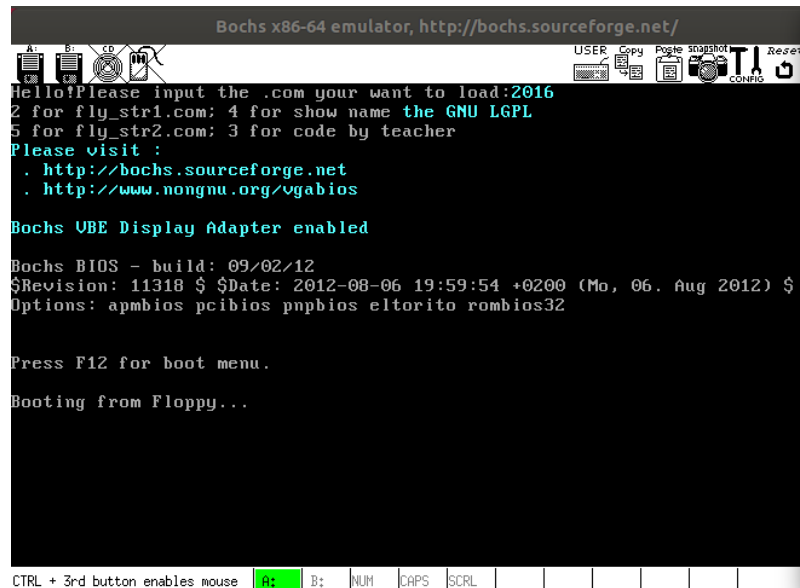
## 用户程序：

- 用户程序的实现与上述过程类似，主要区别在写入软盘这一命令行，将seek的值设为写入的目标扇区号即可,如飞翔字符程序写入第二个扇区，输入：

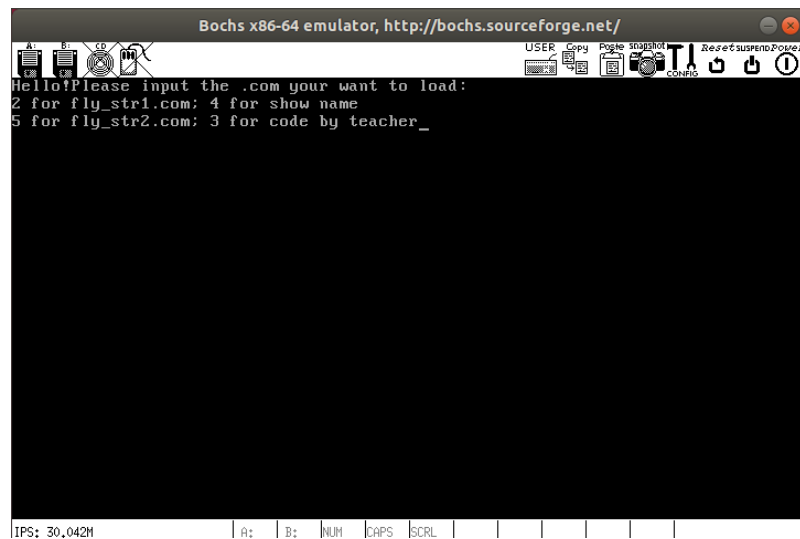
```
1 dd if=fly_str.bin of=a.img seek=1 bs=512  
count=1 conv = notrunc
```

## 虚拟机下运行结果：

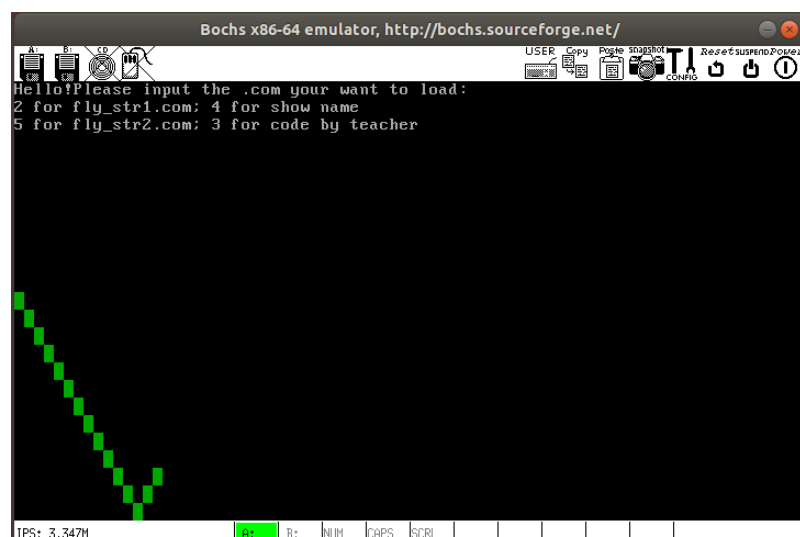
- 系统初始界面：

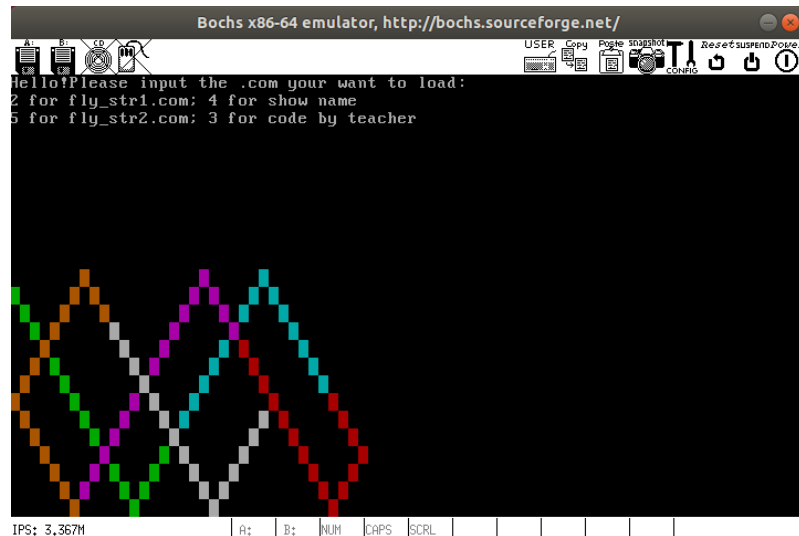


- 用户输入5，调用清屏程序：

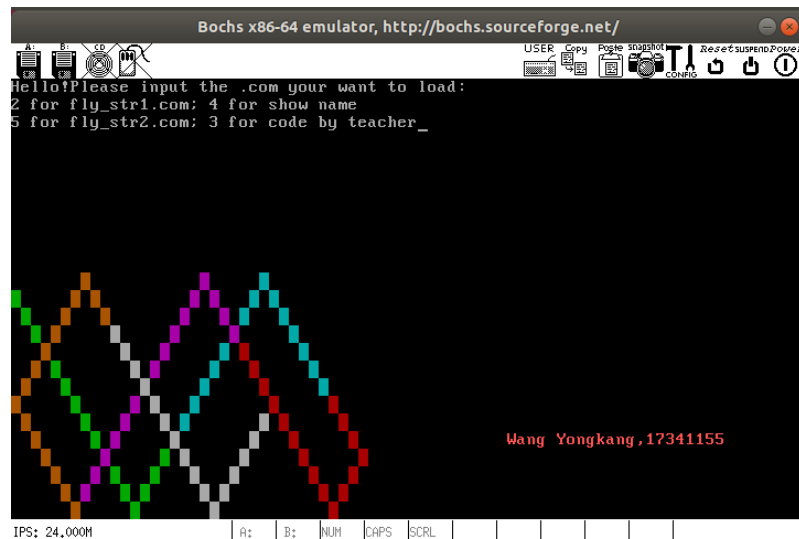


- 用户输入2，调用飞翔字符程序：

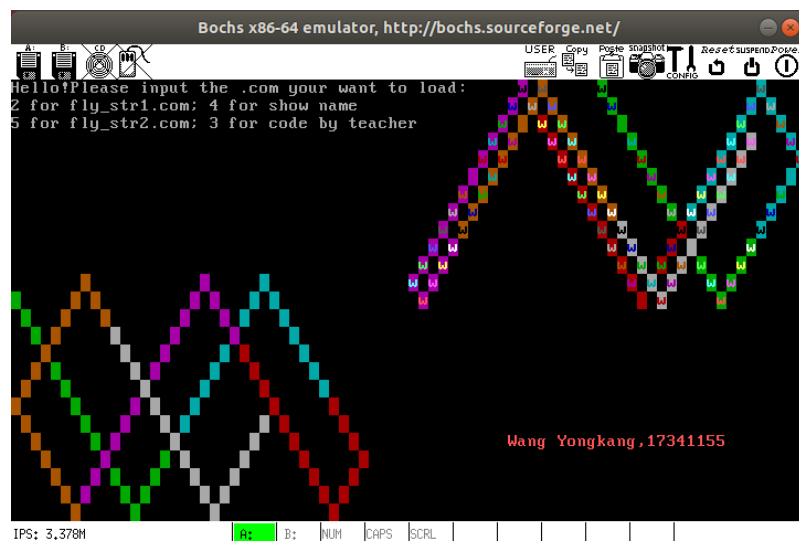




- 用户输入4，调用显示名字、学号程序：

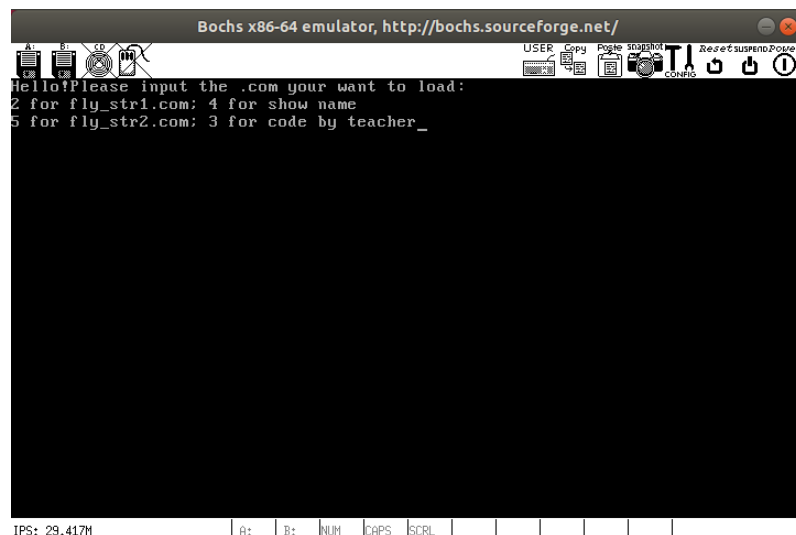


- 用户输入6，调用酷炫字符程序：





- 用户输入5，调用清屏程序：



上述过程中，遇到的问题主要有如何把出现加载到特定的扇区，经过反复查阅资料发现Linux的dd命令可以容易实现这一操作；其次是关于如何从用户程序返回监控程序，经过思考原来只需使用简单的跳转指令到07c00h就行了。

## 五、实验总结

这一次设计监控程序的操作系统实验使我进一步加深了对与操作系统的认识，也是一次在实验一的基础上进行改进的实验，从最初的一台裸机，到现在能够调用程序运行，循序渐进的过程使得对操作系统的认识由浅入深，取得的成果也是令人兴奋。

这一次的实验，对BIOS的认识也是进了一步，它作为一个程序与硬件打交道的计算机内置程序，可以通过它对计算机的硬件进行控制，显示字符通过它的中断着实方便了不少。

其次，对于x86语言进行编程的能力也得到进一步提高，对其语法也是进一步熟悉，与此同时，对于x86系统的CPU架构体系也是进一步理解，这大大的促进了我对于操作系统获得和交出控制权的理解。

希望之后实验能够使操作系统的功能愈发负责、完善，构建出一台真正意义上属于自己的系统！

## 参考文献

- [1] 王爽.《汇编语言(第3版)》[M].清华大学出版社, 2013-9
- [2] 凌应标.“02实验课.ppt”, 中山大学计算机科学系, 2015-3
- [3] <https://blog.csdn.net/longintchar/article/details/70183677>, 2019-3-14