

# OS LAB1 实验报告

141242026@smail.nju.edu.cn

刘驭壬

141242026

## 1.实验环境：

ubuntu

gcc版本：4.8.4

## 2.实验过程

实验初期读了很多遍实验讲义，虽然写过PA，但是还是对框架很不了解，特别是一些底层硬件和Makefile。所以在代码的框架上我参考了很多mit的jos和2012级的nos。在LAB1实验，我大致整合了这两个框架，然后完成了讲义要求的所有内容，游戏方面写了一个双人对战的井字棋，能够相应键盘中断且经测试暂时没有bug。

## 3.游戏编写的说明

对于游戏，我将棋盘分为9块考虑，每块是一个block结构体数组。

在effect.c中我定义了init\_block(),judge\_end()和game\_loop()三个函数，game\_loop是游戏逻辑的主要函数。

在game.c中有定义了init\_game()函数,主要就是时间中断，键盘中断的初始化

在video.c中定义了draw\_border(),draw\_block()和各种清屏函数

(blue\_screen(),white\_screen(),yellow\_screen())

游戏玩法：

QWE

ASD

ZXC

这三行分别对应井字棋的位置。

第一个人为白色，第二个人为蓝色。如果有一方连成三个全屏会刷成该人的颜色。如果平局则全屏变为黄色。

## 3.实验中我遇到的问题

1). in out的内联汇编编写首先出现问题

port前使用的寄存器我原本用了”g”然后死活没用，改成”d”以后可行。

```
in_long(short port) {
    int data;
    asm volatile__(
        "in %1, %0"
        : "=a"(data)
        : "d"(port)
        : "memory"
    );
    return data;
}
```

2) printk的实现。

printk我是用了jos中的框架，但是层层include调用的东西太多，一直出错，后来我就自己写了一下,经确认可用。

(编写游戏的过程没发什么非常难以解决的问题，主要是一些小bug，很容易改过来)

## 4.实验心得

这次实验我觉得非常考察我们自己动手找资料的能力，比如如何从0开始完成讲义内容，这方面助教给了我们很大帮助。我这次主要的参考资料为mit的jos，12级的nos和上学期完成的PA（仔细读过这三个代码后我好像发现了什么，天下代码一大抄啊...），有这三个的帮助再结合助教给的讲义，很容易完成除游戏外的其它部分（很多都有直接的实现，或者在PA中实现过），我要做的主要就是理解这些代码。此外对于自己写Makefile文件，我还不太行。只能照着别人的Makefile小范围自己改改，后续还要再学习一下Makefile。对于内联汇编，得益于这次实验，我基本可以自己写起来。借助这次实验我也加深了对git的了解，较全面地学习了一下git，为了防止git崩坏（曾今的惨痛教训），我在github上同步了git纪录。

注：在提交的文件中有一个note.text 里面是工程中各文件作用的大致介绍