OSlab2实验报告

王亚楠 141220107

邮箱 [wyn1996@hotmail.com](mailto:wyn1996@hotmail.com)

# 实验进度

采取分段方式，已完成除PCB以外的所有实验要求，PCB阶段将在接下来的阶段中完成。

使用的gcc版本：32位

# 试验心得

在实现系统调用阶段没有遇到什么问题，不过通过系统调用中对键盘、串口、显存的封装对系统调用有了更加深刻的认识。

由于我使用的是2012年的版本，所以在分离内核和用户时比较麻烦，进行了整个目录文件的修改，也将Makefile文件全部修改，仿照kernel的生成和链接规则写了game的生成和链接规则。

接下来，我先尝试了分页，直接先通过完善Jos中的pmap.c中的函数，然后写loader函数初始话页面，填写页面，将game的数据和代码段映射到页面上。但是在这个过程中出现了闪频（也就是不断重启）的现象，后来发现是映射过程中偏移量有问题，改了之后，出现了缺页的现象，然后调了很久bug也没有结果，所以迎来了分段时代。

整个分段时间并不是很长，但是在其中遇到了很多问题。

整个分段的流程是这样的：先初始化段表、段寄存器、TSS，然后通过将game的数据和代码映射到自己指定的位置（我是0x3000000），然后获得入口地址，再准备iret所需要的eip，cs，eflags，esp，ss，这样便可以进入用户态，就可以正常的运行我们之前的程序了。

分段过程中我将代码段、数据段、栈全部放到了一个段里，这样很方便，但是可能出现段与段之间相互重叠的现象。而采用段页式的话则可以较好的避免，因为在段页式系统中，为了便于实现地址变换，需配置一个段表寄存器，其中存放段表始址和段表长，进行地址转换时可以先利用段号S，将它与段表长TL进行比较，若S<TL，则未越界，然后再进行地址转换。

下面说说中间遇到的问题：

1. 在读磁盘时直接报13号中断错  
   后来发现，是因为我在中断压栈的时候除了压入中断号，也压入了错误码，所以在返回时esp需要加8，之前加4导致了读磁盘错误。
2. kernel.bin无法被objdump出来查看，会报错  
   经过询问吴楷文，发现是因为kernel所占空间太大，超过了102400，所以超过的数据全部被清除掉了，导致出现问题。后来将kernel的空间扩展到了204800便解决了问题。
3. 解决完读磁盘的问题后还是有13号中断  
   在loader和game中分别加了调试信息，发现程序已经能通过loader进入用户态，但是进入用户态以后马上出现了13号中断错，后来经过调试发现在game中存在一些wait\_interrupt，enable\_interrupt等操作，可能会触发中断错，后来将这段代码注释掉以后就一路畅通，成功实现了分段。

关于讲义中的问题：



\_start=RELOC(entry)，这一行代码是将虚拟地址entry转化为物理地址\_start，然后在低地址运行代码

jmp\*%eax，跳转回高地址

因为内核空间被所有用户空间共享，同时由于user\_supervisor域对页实施的保护，这样做不会带来问题。