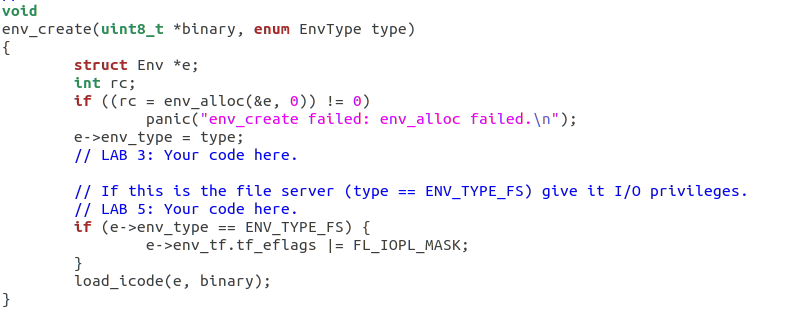
练习1：

在env\_create函数中修改进程的eflag值。

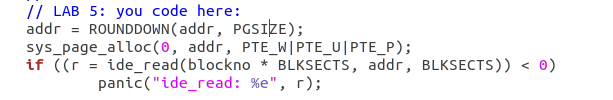


问题1：

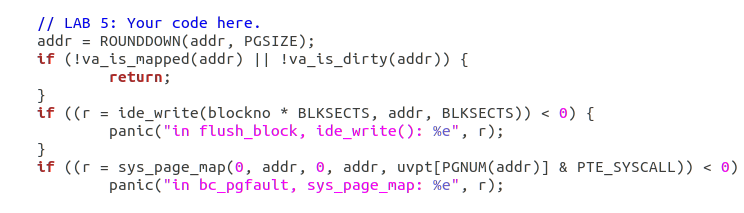
在进程切换时调用了env.pop\_tf函数，其中进行了寄存器的恢复，在iret指令中恢复了eip，cs，eflags等寄存器。

练习2：

bc\_pgfault()是FS进程缺页处理函数，负责将数据从磁盘读取到对应的内存。

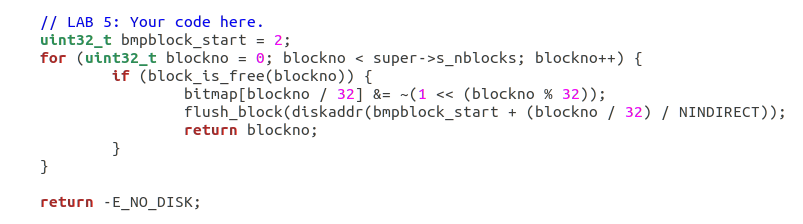


flush\_block()将一个block写入磁盘。flush\_block()不需要做任何操作，如果block没有在内存或者block没有被写过。可以通过PTE的PTE\_D位判断该block有没有被写过。



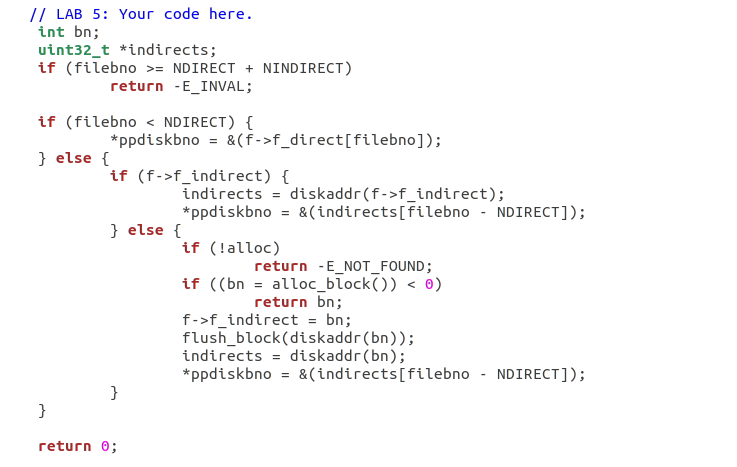
练习3：

实现fs/fs.c中的alloc\_block()，该函数搜索bitmap位数组，返回一个未使用的block，并将其标记为已使用。

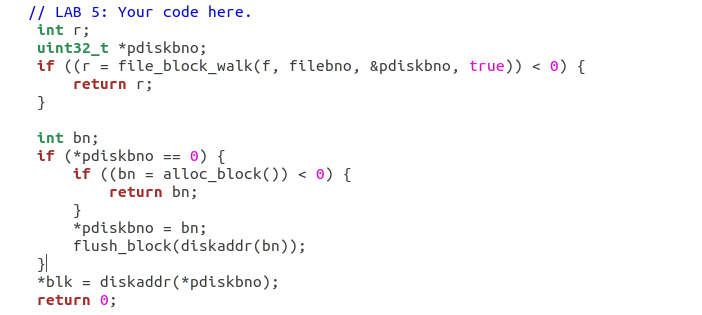


练习4：

file\_block\_walk函数寻找一个文件结构f中的第fileno个块指向的磁盘块编号放入ppdiskbno。

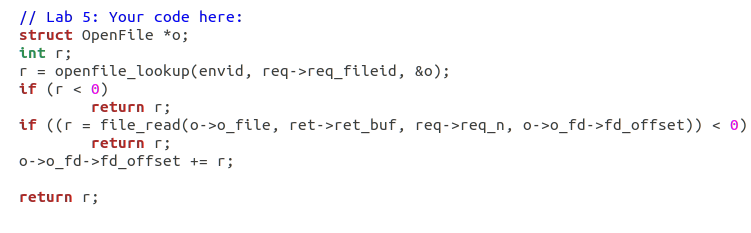


file\_get\_block函数先调用file\_walk\_block函数找到文件中的目标块，然后将其转换为地址空间中的地址赋值给blk。

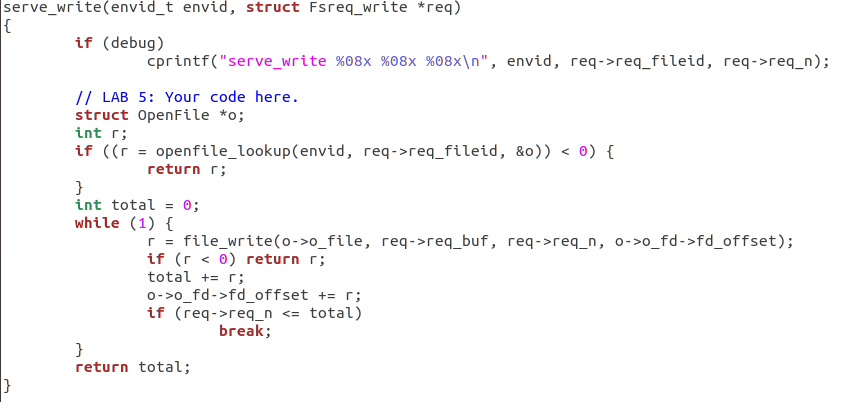


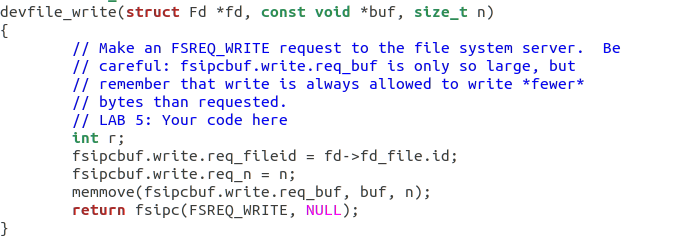
练习5：

实现fs/serv.c中的serve\_read()。这是服务端也就是FS进程中的函数。直接调用更底层的fs/fs.c中的函数来实现。



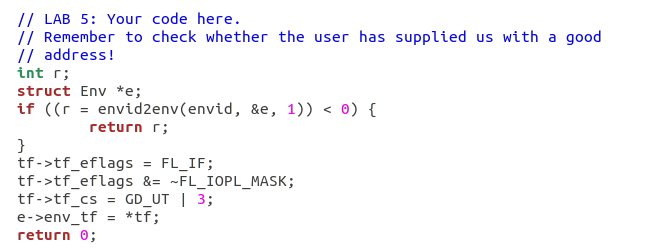
练习6：





练习7：

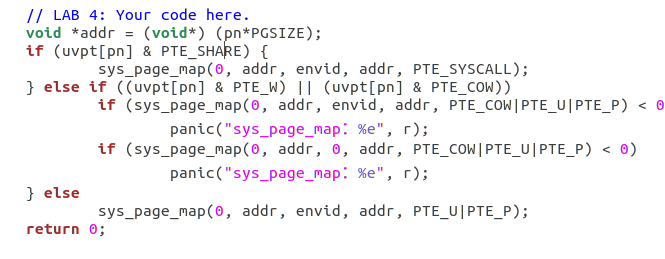
sys\_env\_set\_trapframe函数主要是用来拷贝父进程的寄存器

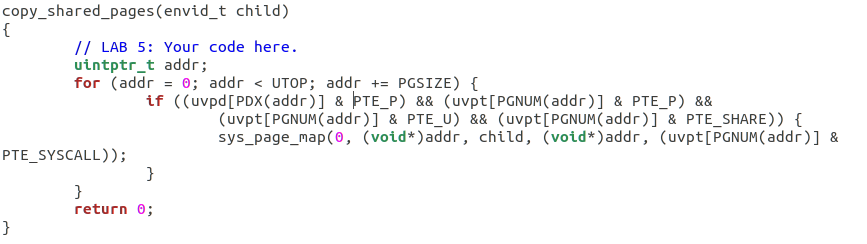




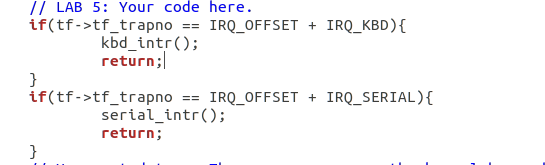
练习8：

修改lib/fork.c中的duppage()，使之正确处理有PTE\_SHARE标志的页表条目。同时实现lib/spawn.c中的copy\_shared\_pages()。



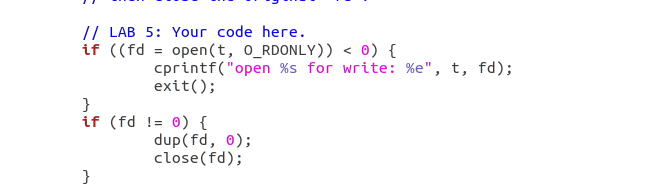


练习9：



练习10：

目前shell还不支持IO重定向，修改user/sh.c，增加IO该功能。



Bug：

在Exercise 7，make grade 时发生了一个错误Protection I/O space: FAIL (2.7s)

跟踪发现错误应该出现在一个 trap 上， 我写的代码没有产生本应出发现是trap.c中 trap\_init\_percpu出现了问题。因为一直没有涉及到I/O， 所以这个Bug也没有浮现。

加入一行代码thiscpu->cpu\_ts.ts\_iomb = sizeof(struct Taskstate);后就正常了。现的 trap。