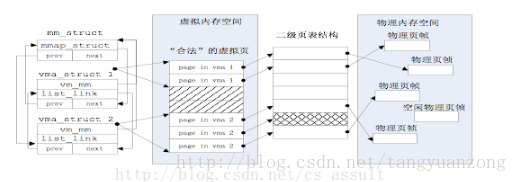
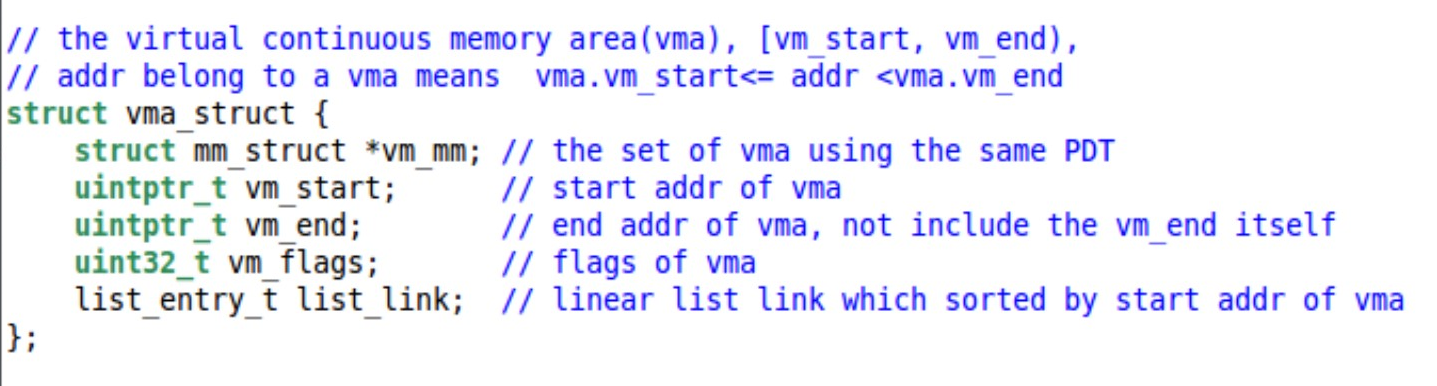
**uCore操作系统Lab3实验报告**

1711291 信息安全 李季航

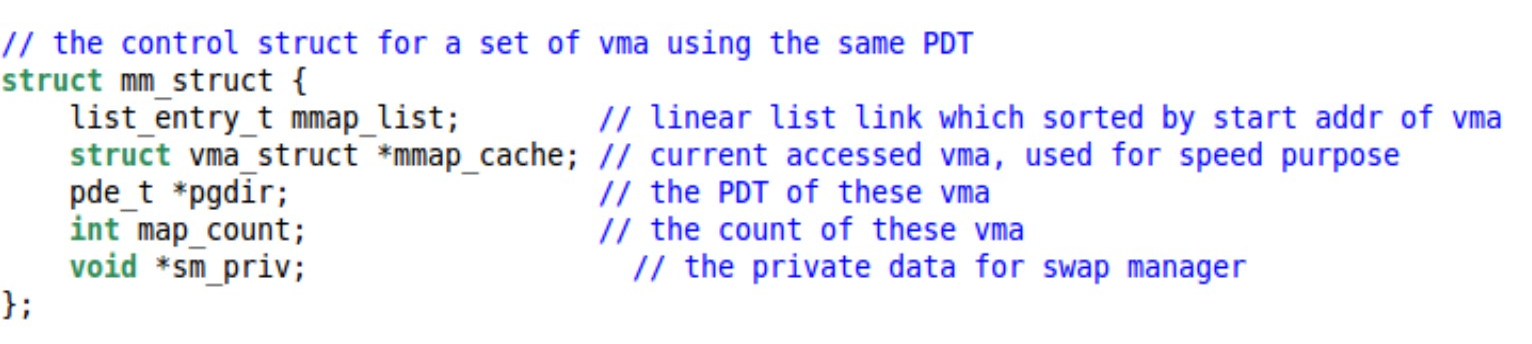
1. **练习1：给未被映射的地址映射上物理页**

**&&练习2：补充完成基于FIFO的页面替换算法**



****

题目中提到的vma是一个用来维护虚拟内存空间的结构体，每个应用程序有一个，其中存有这个应用程序所有的虚拟内存空间的地址位置、一系列标记和维护页表的双向链表。

****

vma上层是一个mm\_struct，管理所有使用相同页目录项的vma。这些vma连接成为双向链表mmap\_list。

1. 逻辑？

缺页分三种：没有权限换出、页表项不存在、页表项映射其他物理页。

权限检查原代码已给出。

通过get\_pte获取物理地址理对应的页表项：页表项不存在时通过create传参=1创建页表项，创建成功返回空页表项后直接分配，创建失败返回NULL；

页表项中有其他物理地址：换出原页，换入新页。

换出规则：在磁盘上分配8个扇区，用PTE描述物理页与磁盘扇区的映射关系（高24位表示此页在磁盘上的起始扇区，PTE\_P空）

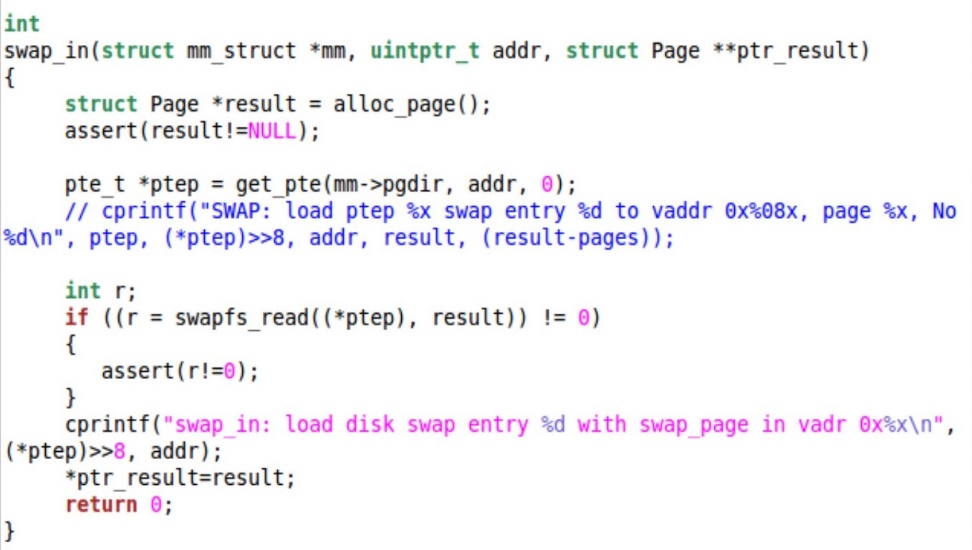
换入规则：swap\_in临时从磁盘中把要换入的页放入内存，page\_insert建立页表到物理地址的映射，把这一页插入。设置swappable和pra\_vaddr记录此物理页对应的虚拟页起始地址。

1. swap\_init\_ok什么时候设置的？

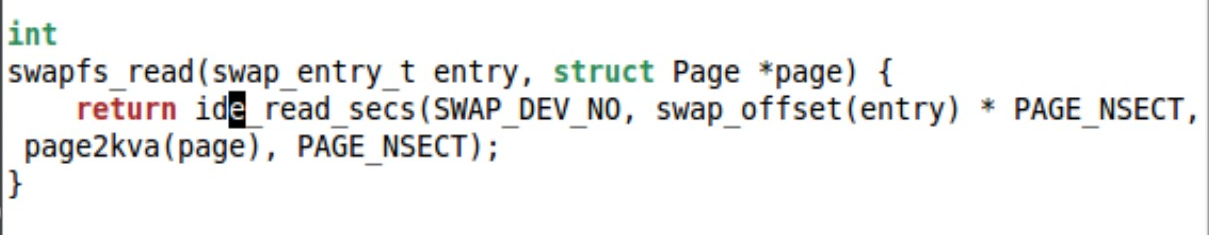
在swap.c初始被置0，也是导致为什么后面qemu进了else分支里出现swap\_init\_ok failed的情况。。

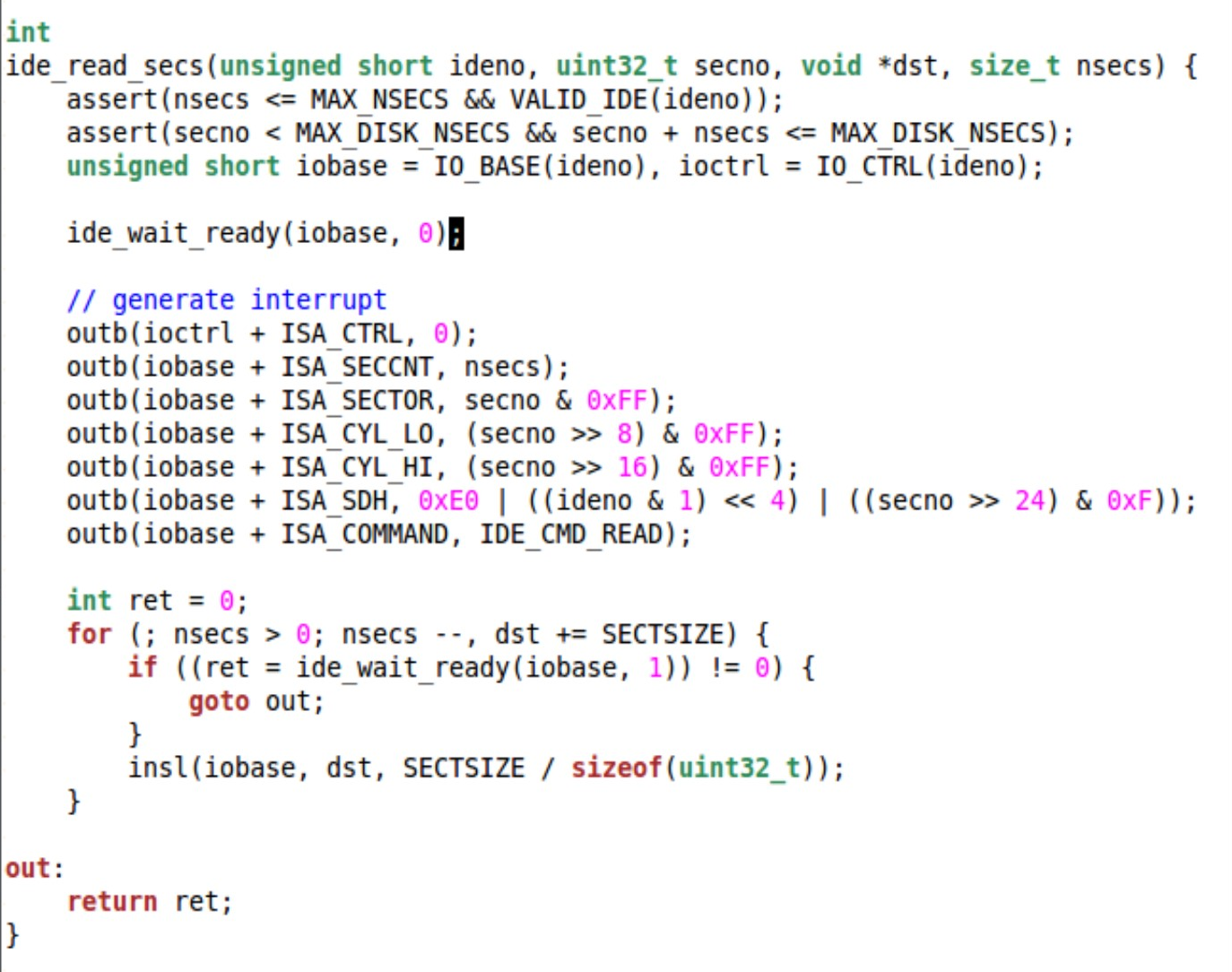
1. swap\_in本体？

开辟一个页空间



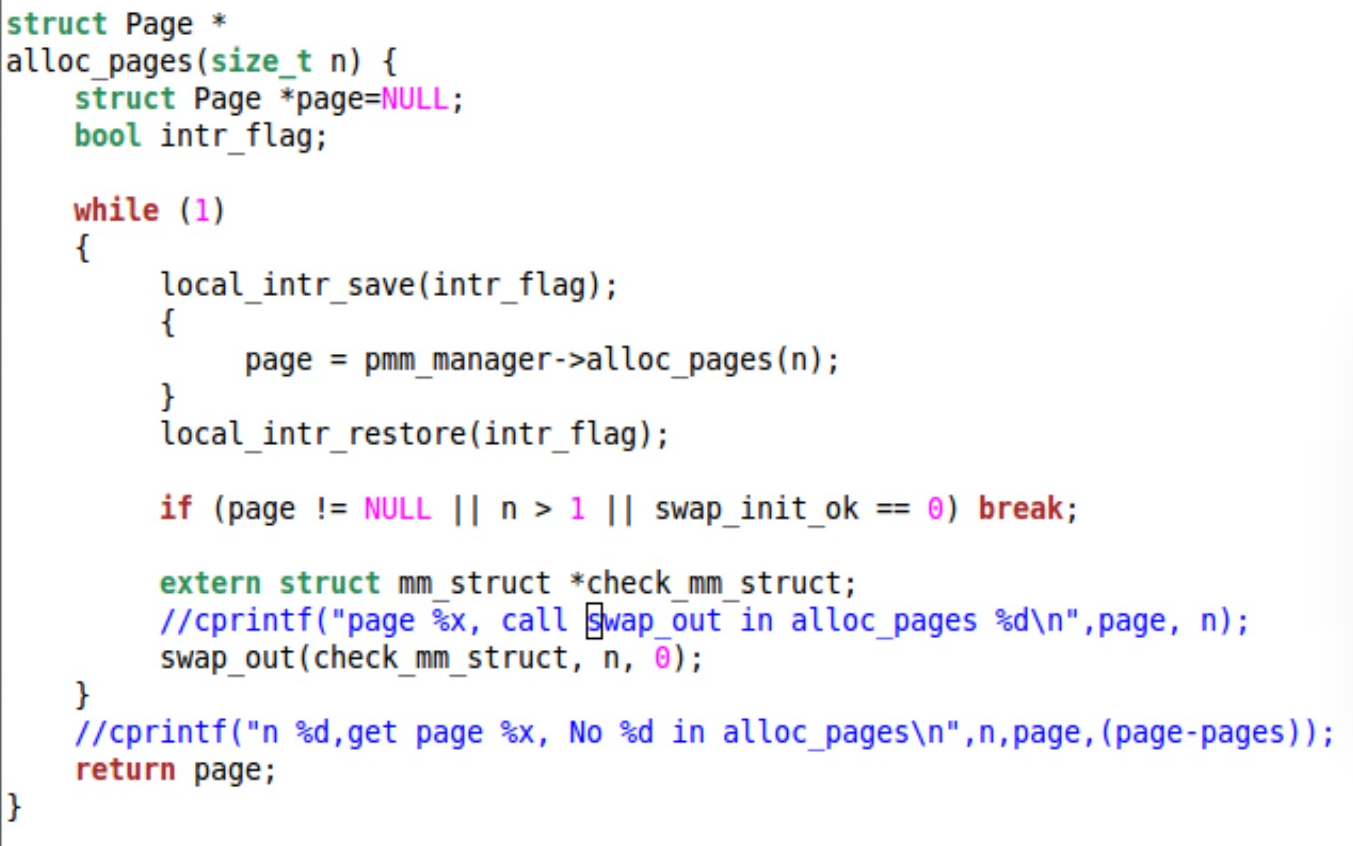
调用swapfs\_read把换入的磁盘部分存到result里

****

****

返回do\_pgfault时把从磁盘读到的存在了page中。

1. swap\_out在哪？



turn out swap\_out发生在分配页却找不到空闲的时候。。所以在do\_pgafult被隐式调用了。。

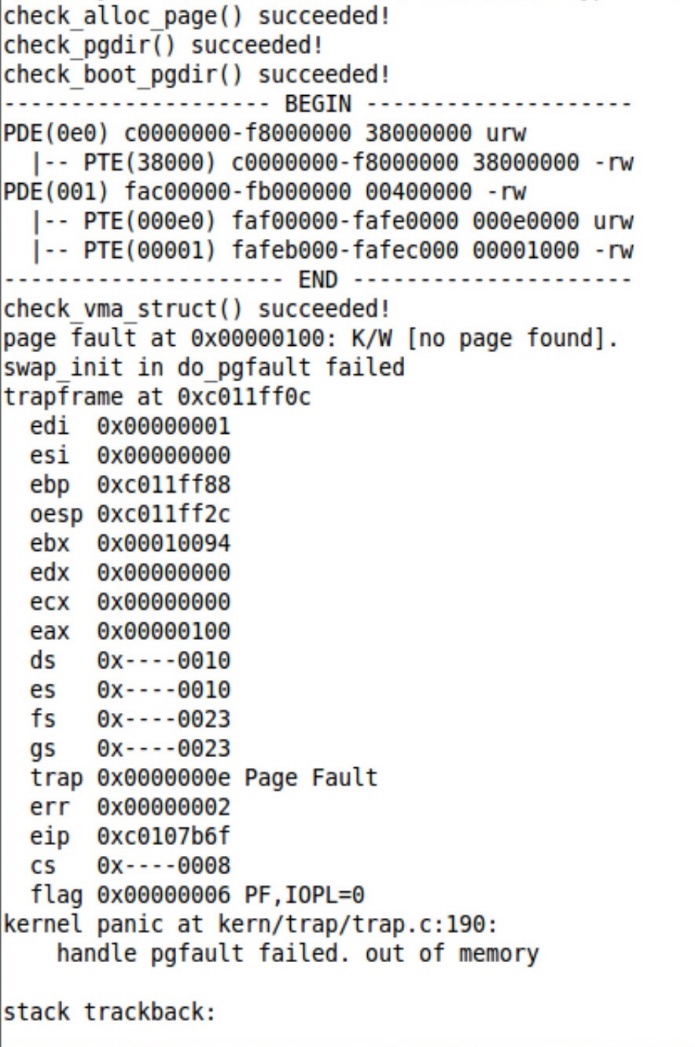
1. make qemu的bug：

最开始的版本：

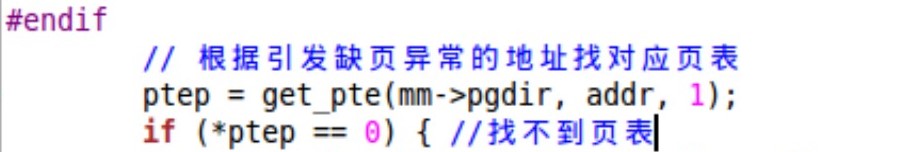




有bug：



在最开始get\_pte之后做了一点小改动却对了：

****

看答案里分别做了ptep==NULL和\*ptep==0的两次if分支判断，原以为这两个的意义是相同的，都是判断是否有已经被映射的别的物理页，表示需要换入的情况。

但回去找了get\_pte的函数明白了之前的理解偏差：get\_pte传的第三个参数1控制create，对于返回NULL和\*ptep==0，前者发生在创建页面失败的情况下，后者代表创建页表项成功故而尚未分配的情况，并不代表找不到页表项。事实上，应该是不存在之前创建过此页表项而此时为空的情况，如果\*ptep==0，则一定是此刻刚创建好的新页表项。

答案万岁！

1. **问题们**
2. PDE和PTE对ucore中实现页替换算法的用处：

PDE作为高级索引，找页面都经过它，还可以用来找swap\_out的页。

页换入时需要将磁盘内容读入内存，换出时将内容写出磁盘。PTE维护权限和虚实地址的映射关系。通过PTE\_P标志指示此时页的位置是处于内存（1）还是磁盘（0），虚实地址映射关系此时不存在，高24位记录写出的磁盘起始位置，为换入换出提供磁盘位置信息。页目录项则是用来索引对应的页表。

1. 如果ucore的缺页服务例程在执行过程中访问内存，出现页访问异常，硬件做哪些事？

这和上一个lab的有啥区别。。？

1. 如果在ucore实现“extended block页替换算法”，给出设计方案。现有的swap\_manager框架是否足以支持在ucore中实现此算法？如果是，给出你的设计方案。如果不是，给出新的扩展和基于此扩展的设计方案。回答：i.需要被换出的页的特征是什么？ii.在ucore中如何判断具有这样特征的页？iii.何时进行换入换出操作？

利用PTE中的PTE\_A和PTE\_D来做换出页的判断。

可以利用timer，每隔一个相等的时间调用一个检查页面状态的函数通过遍历链表循环检查页面，类比指针扫过。

AD为00，则该页面换出。否则，清空A或将AD=01的页面加入写出缓存。

1. **Challenge：实现识别dirty bit的extended clock页替换算法**