OS Lab2 文档

引入部分：

真得很感谢这么细心的讲解git的用法，确实对命令行git很不熟悉，之前都是在图形化下操作的，之前很担心用git会用崩导致lab直接卡在开始的地方。

Part1:

总体是要实现物理地址的alloc和free功能，操作一个free\_list来表示物理页的情况。

boot\_alloc():

一个用来在初始化的时候在物理内存中alloc并拿到相应虚拟地址的函数。实现的方式是讲当前的nextfree作为结果返回，根据n的大小和4k对齐更新nextfree。

mem\_init():

实现了该函数的前面一部分，主要是做了在物理内存里alloc一个page结构来存物理内存的信息。

page\_init():

初始化free\_list，主要是将物理内存中的bios块，io洞和刚刚已经使用的页表和page结构标记为不难再被使用，其他可以使用的内存块放到free\_list中备用。

page\_alloc():

从free\_list中取出一个空page来使用，值得一提的是ref应该由caller更改

page\_free():

将一个ref=0的空页重新放回free\_list中

Part2:

虚拟地址的翻译通过页表来实现，而kernel中将一部分虚拟地址与物理地址直接能以一个固定的差值来映射，这样在还没有页表的时候用加减法就能完成映射。比如页表头和page的结构等。

pgdir\_walk():

通过pgdir来寻找pte结构，如果没有就建立一个，根据注释来实现即可，值得一提的是最后加上PTE（va）的时候要记得转换类型，否则会出难以寻找的问题。原因是void\*和pte\_t \*相加是有区别的。

boot\_map\_region():

直接将连续的va映射到pa上，实现起来直接调用pgdir\_walk() 开辟新的pte然后改写其属性即可。

page\_lookup():

给定va去寻找对应的page是什么，为了remove方便，可以选择将pte也保存。注意物理地址和虚拟地址的转换即可。

page\_remove():

将给定的va映射取消。过程是把ref减一，如果为0就直接free掉，这样就把物理页取消了映射；然后把相应的pte擦掉，这样在页表中也取消了映射；最后将tlb中的记录也清除。

page\_insert():

给定va和物理page建立他们的映射关系。直接搜索页表，找到对应的pte，如果没有就新建一个pte来映射。如果找到了，那么做remove旧页和增加ref以及修改pte的操作。注释中提到了一种corner case，就是要insert的页其实就是原来的页，那么只需要先把ref增加再remove即可，这样由于ref不为0，相应的页不会被free。

Part3:

mem\_init():

用boot\_map\_region和boot\_map\_region\_large映射kernel虚拟内存的权限和对应物理内存。在加载cr3前打开cr4中的size extension位。有一个疑问是kernelstack的guard page要如何设置呢？我现在是没有映射任何东西，但是感觉这样似乎不够。

boot\_map\_region\_large()：

和boot\_map\_region差不多，pgsize改成ptsize的页，然后打开pde的page extension bit。

Challenge:

1. chunk

/\*

\* My core method: Use a blank\_list to save the unused block explicitly

\*

\* My mem struct: # One |...| represents one word #

\* |BLACK\_LISTHEAD|PADDING|PADDING|PROLOGUE HD|

\* |PROLOGUE FT|.........DATE........|EDILOGUR|

\*

\* My data struct:

\* (ALLOCATED) |HEADER|.....|FOOTER|

\* (BLANK) |HEADER|......|PREV|NEXT|FOOTER|

\*

\* Init: HEADLIST = NULL

\*

\* Alloc: Search the blank\_list for the proper block(first-fit)

\* if no one fit, expand the mem.

\* Then, remove the allocated block from list

\*

\* Free: Set the alloc bits to 0, then insert the blank block

\* into blank\_list

\*/

一个allocated chunk list存所有chunk

一个 blank chunk list存所有blank

具体initial alloc free策略如上所述