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