Lab2 文档

1. 物理页管理

   boot_alloc()函数判断当前是否还能分配要求的页面数量，然后将表示空闲页面开始地址的静态变量 nextfree 赋给返回值，并将 nextfree 增加分配的内存数量

   mem_init()仿照 kern_pgdir 的初始化方法，初始化 pages

   page_init()是一个 for 循环，需要在其中判断页面类型初始化 pages[i]的属性，pp_link 的值是指向前一个空闲页的指针（反的链表），链表的头指针是 page_free_list

   page_alloc()从 page_free_list 得到空闲的页面，更新 page_free_list

   page_free()将当前页面插入到空闲链表的头部

2. 虚拟内存

   pgdir_walk()根据先寻找 va 对应的 PDE 的位置，进一步得到对应 PT 的位置，最后返回页面对应的 PTE 的指针。如果没有对应 PT，根据参数决定是否创建新的 PT，然后设置 PDE 的权限，返回页面对应的 PTE 的指针

   boot_map_region()是一个 for 循环，在指定的虚拟地址范围为对应的 PTE 条目赋值，实现映射

   page_lookup()调用 pgdir_walk()返回对应 PTE，然后调用 pa2page()返回对应的页面

   page_remove()先调用 page_lookup()检查页面，然后减少 refcnt，清空 PTE 条目，设置 tlb

   page_insert()调用 pgdir_walk()返回对应 PTE，判断是否新插入的页面的 PA 和当前映射的 PA 是否相同，不同则 page_remove()，删去原来的映射。最后增加对应页面的 refcnt，更新 PTE 内容

3. 内核地址空间

   mem_init()调用 boot_map_region()根据注释要求映射虚拟和物理页面，注意分配 KERNBASE 以上的物理页面需要将大小指定为(0xffffffff-KERNBASE)+1

   boot_map_region_large()是一个 for 循环，页表只有一级，把每个 PDE 的内容设置，实现映射