

## Ex0

Merge files:

kdebug.c

trap.c

## Ex1 实现 first-fit

default\_alloc\_pages:

该函数用于分配 pages, 首先判断空闲页大小是否大于所需页块大小。如果小于空闲页大小, 遍历整个空闲链表, 找到合适的空闲页则重新设置标志位, 然后从空闲链表中删除此页。

default\_free\_pages:

释放已经使用完的页, 并合并到空闲链表中。在空闲链表中查找合适的位置并插入。

改进空间:

保存空闲块列表时保存了全部空闲页, 而对于连续空闲块可以仅仅保存第一页和长度。

## Ex2 实现寻找虚拟地址对应的页表

对照注释编码即可。

PDE 的含义:

高 20 位为该 PDE 对应的页表起始位置 (页基址)

9-11 位为保留位, 可以被用户使用

7 位设置页大小

6 位为上次清零过后该页是否被写

5 位为上次清零过后该页是否被读写

4 位为该页是否缓存

3 位为缓存策略: 1/Write Through, 0/Write Back

2 位为访问需要的特权级

1 位是否可写

0 位存在位

PTE 的含义:

高 20 位为 PTE 指向物理页的基址

低 12 位与 PDE 一致

出现页访问异常:

# 将发生错误的地址保存在 CR2

# 中断栈中压入 EFLAGS/CS/EIP, 以及 ERROR CODE, 若是在用户态发生 page fault, 还需要压入 SS/ESP 并切换至内核态

# 查询中断描述符表, 查到对应的 ISR 并执行

## Ex3 释放页、取消映射

按照注释编码即可。

- 数据结构 **Page** 的全局变量（其实是一个数组）的每一项与页表中的页目录项和页表项有无对应关系？如果有，其对应关系是啥？

有对应关系。可以通过高 20 位物理页号，即物理页的物理地址，获取到 Page 数组中的项，再根据 Page 与页大小不同，计算得到偏移量。

- 如果希望虚拟地址与物理地址相等，则需要如何修改 **lab2**，完成此事？**鼓励**通过编程来具体完成这个问题

更新 gdt 与段映射，使得  $VA = \text{Linear address} - \text{offset}$ ，offset 为 0xc0000000 时  $VA=PA$

make grade pass

```
moocos-> make grade
Check PMM:                               (2.8s)
  -check pmm:                             OK
  -check page table:                       OK
  -check ticks:                           OK
Total Score: 50/50
[~/moocos/ucore lab/labcodes/lab2]
```