操作系统原理实验 （Lab 3）

# 实验目的：

# 理解基于段页式内存地址的转换机制

# 理解页表的建立和使用方法

# 理解物理内存的管理方法

# 实验内容：

练习0：填写已有实验

本实验依赖实验1。请把你做的实验1的代码填入本实验中代码中有“LAB1”的注释相应部分。提示：可采用diff和patch工具进行半自动的合并（merge），也可用一些图形化的比较/merge工具来手动合并，比如meld，eclipse中的diff/merge工具，understand中的diff/merge工具等。

练习1：实现 first-fit 连续物理内存分配算法（需要编程）

在实现first fit 内存分配算法的回收函数时，要考虑地址连续的空闲块之间的合并操作。提示:在建立空闲页块链表时，需要按照空闲页块起始地址来排序，形成一个有序的链表。可能会修改default\_pmm.c中的default\_init，default\_init\_memmap，default\_alloc\_pages，

default\_free\_pages等相关函数。请仔细查看和理解default\_pmm.c中的注释。

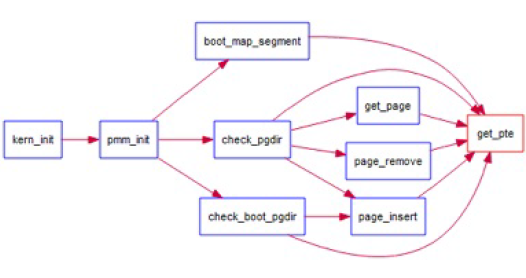
请在实验报告中简要说明你的设计实现过程。请回答如下问题：你的first fit算法是否有进一步的改进空间

练习2：实现寻找虚拟地址对应的页表项（需要编程）

通过设置页表和对应的页表项，可建立虚拟内存地址和物理内存地址的对应关系。其中的

get\_pte函数是设置页表项环节中的一个重要步骤。此函数找到一个虚地址对应的二级页表项的内核虚地址，如果此二级页表项不存在，则分配一个包含此项的二级页表。本练习需要补全get\_pte函数 in kern/mm/pmm.c，实现其功能。请仔细查看和理解get\_pte函数中的注释。

get\_pte函数的调用关系图如下所示：



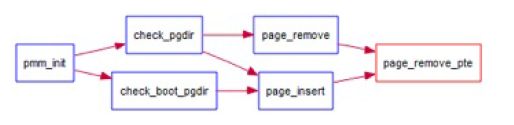
请在实验报告中简要说明你的设计实现过程。请回答如下问题：

请描述页目录项（Page Directory Entry）和页表项（Page Table Entry）中每个组成部分的含义以及对ucore而言的潜在用处。

如果ucore执行过程中访问内存，出现了页访问异常，请问硬件要做哪些事情？

练习3：释放某虚地址所在的页并取消对应二级页表项的映射（需要编程）

当释放一个包含某虚地址的物理内存页时，需要让对应此物理内存页的管理数据结构Page做相关的清除处理，使得此物理内存页成为空闲；另外还需把表示虚地址与物理地址对应关系的二级页表项清除。请仔细查看和理解page\_remove\_pte函数中的注释。为此，需要补全在kern/mm/pmm.c中的page\_remove\_pte函数。page\_remove\_pte函数的调用关系图如下所示：



作业提交要求：

1. 提交至<https://easyhpc.net/course/110>
2. 提交报告和修改后的kernel源代码压缩文件 .zip或者.rar，源代码要求编译后在能在虚拟机上运行
3. 截止日期为2021年5月14日23:59