0006

0005
C.RPL
（彩蛋：学堂在线视频后练习题这题答案错误，误选 A.　DPL。）
（答案出处：操作系统 ucore 实验指导书 pdf 版　p112）
——by：群主(289212182)
——

0004
C，因为根据课本上归并排序的定义，0＜＝lo＜hi＜＝size。如果要把四个元素全部进行排序，应该从 0 开始，hi 应该指向最后一个元素之后的位置，且 hi-lo 刚好等于元素个数。
（答案出处：邓俊辉老师教材，第二章关于向量 Vector 类的定义代码、mergeSort 归并排序实现代码、群内同学优质回复）
——by：可爱至极小鲁酱<lylylulu@qq.com>
——

0003
截取 sign.c 文件中的部分源码

：

```
char buf[512];                                          //定义 buf 数组

memset(buf, 0, sizeof(buf));

                                                                  // 把 buf 数组的最后两位置为 0x55, 0xAA

buf[510] = 0x55;

buf[511] = 0xAA;

FILE *ofp = fopen(argv[2], "wb+");

size = fwrite(buf, 1, 512, ofp);

  if (size != 512) {                                    //大小为 512 字节

       fprintf(stderr, "write '%s' error,

                    size is %d.\n", argv[2], size);
```

```
return -1;
}
```

主引导扇区的规则如下：

1 大小为 512 字节

2 多余的空间填

3 第 510 个（倒数第二个）字节是 0x55，

4 第 511 个（倒数第一个）字节是 0xAA。
（答案出处：操作系统 ucore 实验指导书）
——by：群主(289212182)
——

0002
当字符集规模较小时，单次比对的成功概率较高，蛮力算法的效率较低。此时，KMP 算法稳定的线性复杂度，更能体现出优势。当字符集规模较大时 KMP 算大与蛮力算法相当。

（答案出处：数据结构习题解析第三版(邓俊辉老师)_p218_11-10）
——by：群主(289212182)
——

0001
讨论 1：物理地址（——by：Crowerly）

讨论 2：[pic]这是 entry.s 的代码；这是系统头一次初始化页目录表和二级页表的地方；[pic]这是我在 pmm_init 中的测试 ；最新的代码对于 entry.s 有所修改 但是基本一样；@nullptr ； 二级页表的物理地址基本是通过循环迭代算出来的；那个_boot_pt1 就是代表了一个二级页表的地址；抱歉 是我的问题 他调用了 realloc 就是物理地址（——by：&你的心 link 我的心）

讨论 3：@&你的心 link 我的心 同学 页目录如果存的是虚拟地址，那还怎么定位二级页表的页啊（——by：Crowerly）

讨论 4★：“CPU 可以看到的地址都是虚拟地址，经过 MMU 后才会有物理地址。@Crowerly 定位二级页表的事由 MMU 做。”（——by：向老师）

讨论 5：@向老师 老师 页表就是用来给 MMU 做地址翻译的啊，如果一级页表里头存的是虚拟地址，MMU 就无法定位二级页表的位置了吧；我再回去确认一下吧；（——by：Crowerly）

讨论 6：其实 你们说的就是对的；代码是通过 realloc 一个简单的减法；实现了物理地址和虚拟地址的转换；老师的意思是强调 CPU 拿到的地址是虚拟地址；我们讨论的地址是 mmu 看到的地址；页表也是由 mmu 来实现的；不冲突；我声明我观点的错误 因为没有看到 realloc；页目录表和二级页表都是存储的物理地址包括 cr3 代码中均有体现；所站视角不同；（——by：&你的心 link 我的心）

讨论 7：@某普通大学的数学汪　仔细捋了捋代码，确实是物理地址，书上也写了是物理地址（——by：nullptr）

讨论 8★：也就是说根据线性地址生成最终物理地址的工作由 MMU 完成。cpu 拿到的第一手地址都是虚拟地址，由分段机制生成线性地址，若不启动分页则线性地址就是最终物理地址，若启用分页机制则再对线性地址进行加工。所以页表的地址是物理地址还是虚拟地址哇？（——by：群主 289212182）

讨论 9：cr3 有页目录表物理地址，然后查找到存放的页表物理地址，然后从页表查到物理页号和 offset 拼起来，应该是这样吧?（——by：null）

讨论 10★：我突然想起来一件事情；咱们访问必须是 32 位地址。页目录项给的所谓"物理地址"是这种可以直接拿来不做任何变化访存的地址吗？；如果不是，mmu 必须对它进行变换，那所谓的"物理地址"也就不是真正的物理地址鸭？；能这样理解吗？轻喷。（——by：群主 289212182）

讨论 11：页目录存的是物理页框号；确实不是真的物理地址；真的物理地址需要做 12 位左移位。（——by：Crowerly）

讨论 12★：是吧？我觉得所有的访存工作都是 mmu 在做，cpu 并不知情，它俩谁也不认识谁；个人粗俗理解（——by：群主 289212182）

讨论 13：…（若干）

讨论 14★：“我注意到大家对虚拟地址和物理地址的讨论。这是操作系统课的重要概念，需要仔细理解代码才能有准确的了解。有必要进行深入的讨论。
我先说说我的理解。
1. 在保护模式下 CPU 可以看到的地址都是虚拟地址，经过 MMU 后才会有物理地址。定位二级页表的事由 MMU 做。所以，CPU 不能直接用物理地址来访问内存，而必须使用虚拟地址来访问。这时才有，CPU 要修改页表项内容时，也是通过虚拟地址来访问的。
2. 在 X86-32 CPU 上，物理地址可能不是 32 位的。如在使用物理地址扩展（PAE）时，物理地址会是 36 位，使用 4KB 页面大小时物理页号也就变成了 24 位，于是一个页表项就占了 8 字节。
3. CR3 寄存器中保存的是页目录的起始物理地址，CPU 只在地址转换中使用它的内容。
请同学们继续置疑和修正描述，并补充相关的代码例证。希望有同学来整理大家的交流结果，并放到 Piazza 上，以方便以后的同学。谁有兴趣来做此事？”（——by：向老师）
——