

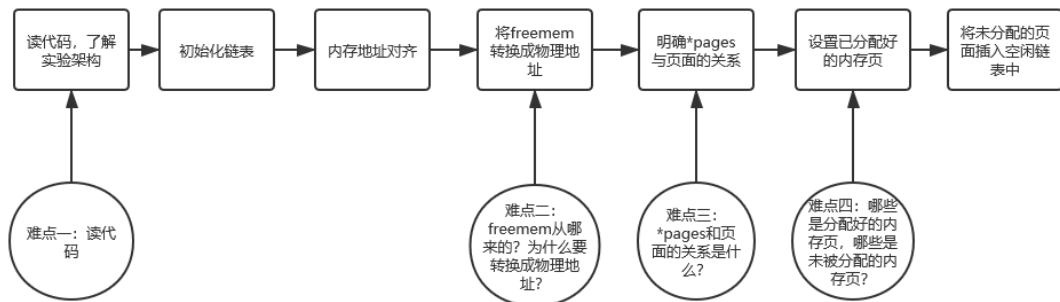
一. 思考题

1. 好处: 可直接通过 CPU 提供的虚拟地址来访问, 不需要经过地址转换, 可提高效率
坏处: 每个进程都有一个独立的虚拟地址空间, 如果两个进程发出的虚拟地址相同就会发生 Cache 冲突, 就会出现访问非自己进程所有的数据的情况, 出现错误。
而通过物理地址访问 Cache 就会保证地址唯一, 不会破坏安全性。
2. 可以设置在 CPU 发出虚拟地址之后查 TLB 和 MMU 执行正常的页表查询工作同时进行, 如果 TLB 命中, 则 Cache 接收 TLB 的发出的物理地址; 如果 TLB 未命中, 则接收 MMU 发出的物理地址。
3. 虚拟地址
4. 原理: 链表为双向链表, 头节点为一个特殊的结点, 内部只有记录之后结点的指针, 其余的结点数据域有记录页面使用情况的 pp_ref 和一个包含两个指针*le_next 和 **le_prev 的结构体 pp_link, *le_next 是指向下一个结点的地址的指针, **le_prev 是一个指向前一个结点的*le_next 的指针。
好处: 将对链表的操作进行封装, 减少的代码的冗余, 也减少了 bug; 通过使用双重指针方便实现对链表的插入和删除操作, 例如 *(elm)->field.le_prev = LIST_NEXT((elm),field) 和 “ LIST_NEXT((elm),field)->field.le_prev = (elm)->field.le_prev;”两条语句可实现对链表的删除操作, 简单快捷不易出错
5. 使用 do{...}while(0)构造后的宏定义不会受到大括号、分号等的影响, 总是会按期望的方式调用运行。
其他优点: 避免空宏引起的警告; 代替 goto 语句; 将语句封装, 变成函数的形式。
6. 物理内存页在内存的 0x80000000 到 0x84000000 之间
*Pages 是一个 Page 类型的指针, 是一个虚拟地址, 通过宏定义 PADDR(kva)将虚拟地址转换成物理地址 (将最高位清零即可)
7. C
8. b 指向的是一个虚拟地址
9. 页目录起始地址为 0xC0300000
10. 不能由虚拟地址直接得到对应的页表项虚拟地址
又必要查目录, 页表是存储在内存当中的, 需要通过物理地址来找到相对应的页表项
11. 先将 CP0_ENTRYHI 的值写入 k1(保护现场?), 将 a0 寄存器的值写入 CP0_ENTRYHI, 接下来查找 TLB, 将 CP0_INDEX 写入 k0 寄存器中, 如果 k0 寄存器中的值小于 0, 则说明 TLB 未命中, 跳转到 NOFOUND, 将 k1 寄存器中的值写回 CP0_ENTRYHI(回复现场?)
4 条 nop 指令的作用是给查询 TLE 的操作留有时间, 避免错误发生
12. 因为我们先执行 page_insert 函数, 其中会调用 tlb_invalidate 函数, 而 tlb_invalidate 函数又会调用 tlb_out 函数, 也就是说在插入页的时候我们就会将 va 对应的页项从 tlb 中删除, 所以在向*va 处写入数据的时候, 我们会出发 tlb miss, 引起异常, 没有重填机制的 lab2 就会在这里陷入死循环
13. 当 CR4 的 PSE 置 1 时, 内存当中的每个页面大小为 4MB。相当于将我们的一个二级页表的 1024 个页表项所对应的物理页进行组合, 每一页 4MB, 其虚拟地址的高 10 位用来查找页目录, 后 22 位做为页内偏移。

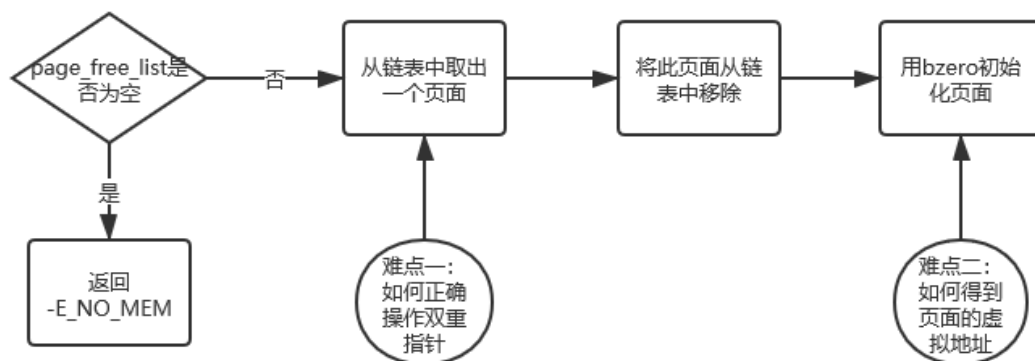
二. 实验难点

在本次实验中，最难的就是理解整个物理内存的分配和管理过程，其中虚拟地址、物理地址的识别和转换是关键部分。在此次 lab2 的 8 个 exercises 中我认为最难的也是我花费时间最多的是 2.3 和 2.4 完成 `page_init` 和 `page_alloc` 函数，因为 2.1 和 2.2 主要是对指针和变量进行操作，难度不大，到了 2.3 和 2.4 不仅需要了解虚拟地址和物理地址之间的关系、同时也要对整个实验的架构有一定的了解，就是要将大部分的代码读一遍，任务量比较重；而有了前面的基础，后面的 4 个 exercises 相对来说就比较轻松。

`page_init` 函数：



`page_alloc` 函数：



三. 感想和体会

此次的 lab2 需要填写的代码并不多，但是需要花费较多的时间去理解二级页表机制，其中我认为最关键的就是识别和转换虚拟地址和物理地址。在这次的实验中，我体会到了运用好宏对于代码的简洁性和正确性的好处，同时阅读宏对于理解整体的实验架构也有很大的帮助；最重要的是了解到了操作系统对内存的管理机制，虽然这个机制还未完全建立（缺少页面调度等功能），但随时实验的进行，将进一步理解详细完整的内存管理机制。

四. Lab2-extra

此次的 lab2-extra 主要是实现从高地址申请物理内存，并对物理页的不同状态进行输出。由于在之前的 lab2 中对虚拟地址和物理地址之间的转换有一定的了解，所有没有花费很多的时间。但是在之前没有理解透彻物理内存的分布，导致物理内存的下限不清晰，和同学讨论知道了就是“end”，所以在之后的实验中要多关注这些特殊标志的作用。