

1. 习题 1

寄存器, 缓存速度快, 但成本高, 硬盘成本较低但速度慢,
采用存储层级结构可以缓解内存墙同时获得较多的容量, 提高性价比,
减少成本。

2. 习题 2

过大的页: 浪费内存空间

过小的页: 页表项过多, 页表巨大, 地址翻译更复杂

3. 习题 3

1) 7: dirty 表明该页是否被改写 (是否被写脏)

6: accessed 表明该页是否可访问

5: global 全局页面标识, 当页可供多个进程共享

4: user 用户模式是否可访问

3, 2, 1: 可执行, 可写, 可读

0: valid 表明物理页在内存中是否分配好

2) 用户进程可以将表项修改为指向其他进程的内存位置, 用户进程间可互相干扰,
权限机制将形同虚设。

3) 下一级页表的指针

4. 习题 4

1) 作用: 对物理地址的访问权限进行检查, 判定CPU是否具备该地址的
读/写/执行权限

2) A位: 表项的地址匹配模式

00: off, 无效表项 01: TOR, 使用相邻表项的地址作为匹配区间

10: NA4, 区间大小为4字节的匹配模式

11: NAPOT, 区间大小为2的幂次方的匹配模式

L位: 表项的 Lock 使能位

- 0: 机器模式访问都将成功, 系统用户模式根据 R/W/X 判定
1: 表项被锁住, 无法修改相关表项, 当配置 TOR 模式, 其前一个表项的地址寄存器也无法被修改, 所有模式都需要根据 R/W/X 判定是否访问成功

5. 习题 5

1) $\frac{2^{64}}{4 \times 2^{10}} \times 8 = 2^{55} \text{ B} = 32768 \text{ TB}$

2) $\frac{2^{48}}{4 \times 2^{10}} \times 8 = 2^{39} \text{ B} = 512 \text{ GB}$

3) 多级页表能随着进程占用内存空间的增大对应增多属于该进程的页表数目, 而当进程占用内存空间很小时, 页表数目也很少, 以此减少页表占用的内存空间。