

1. 存储系统的层次化主要是为了提高性价比。由于不同存储技术在技术指标和成本上的差异，存储采用多层的体系，其中最底层为持久化存储层，速度最慢、容量最大、成本最低；越往上速度越快、容量越低、成本越高。这样就使得整体成本接近底层，性能接近最上层，容量和持久性等同持久化存储层。

2. 过大的页：当进程只需要使用很小的内存时，使用大页就会造成大量内存浪费，同时使内存碎片增加。

过小的页：虽然减少了内存碎片，但造成每个进程占用较多的页面，从而导致页表过大，内存利用率降低。

3. (1) D-Dirty：D位表明该页已经被写；为0时表示未被写/不可写。若此时仍对该页进行写操作会触发 Page Fault (Store) 异常。

A-Accessed：A为1时，表明该页可访问；为0时不可访问，否则会触发 Page Fault 对应访问类型异常。

G为1时

G-Global；全局页面标识，当前页面可供多进程共享；为0时非共享，进程 ASID私有
U-User；U位为1时用户模式可访问；为0时用户模式不可访问，否则出现 Page Fault 异常

X W R 分别为可执行、可写、可读标识：都是有效，全为0时，指针指向下一级页表

V-Valid：表明物理页在内存中是否被分配好，V为1时表明已分配好，访问一个V=0
的页面，则将吊出发 Page Fault 异常

(2) 如果用户能自由修改自己的页表，可能会使一些如用户模式不可访问或者不能写的页表
产生 Page Fault 异常

(3) XWR全为0表示 指针指向下一级页表

判断当前工作模式下 CPU 对特定物理地址的可读、可写、可执行权限

4. 解释：(1) X：X为1表明表项匹配地址可执行，为0则不可执行

W：W为1表明 PMP 表项匹配地址可写，为0则不可写

R：R为1表明表项匹配地址可读，为0则不可读

(2) L：Look 使能位；为0则机器模式访问都成功，系统/用户模式则由 R/W/X 判定

为1则表项被锁住，无法对相关表项进行修改

当配置了 DR 页表，且前一个表项的地址寄存器也无法修改
所有模式根据 R/W/X 判定是否可访问

A. 表项的地址匹配模式：00：OFF，无效表项；01：TOI，使用相邻表项地址作为匹配区间

10：NAT，区间大小为字节的匹配；11：MAPT，区间大小为2的幂次匹配，至少

5. 答案：(1) 页内索引占12位，剩15位，所以需要 $2^{12} \times 8$ 字节，即 2^{35} 字节空间

(2) 页内索引占12位，剩36位，所以要 $2^{36} \times 8 = 2^{39}$ 字节空间

(3) 因为多级页表是一对多的关系，每一页引出的内容用来索引下一级的地址，
且省却了大量未映射的页表项，只保留如第一级页表，1个第二级或第
三级在内存中，从而大幅减少了空间