

4.1 现代计算机需要存储层级的原因是为了在速度和容量之间找到一个平衡点。计算机需要快速访问数据和指令，同时也需要存储大量的数据，例如文档、图片、音频、视频等。如果所有数据都存储在相同的存储设备中，读写速度会变慢，而且成本也会变高。为了解决这些问题，计算机设计了存储层级，将数据分散在不同类型的存储设备中。

4.2 过大的页：① 当一个进程只需要一小部分某个页时，整个页都会被调入内存，这可能会浪费大量的内存空间和资源，造成内部碎片。② 当一个进程需要使用大量的页时，页表会变得很大，导致查找和修改页表非常慢。

过小的页：① 由于页面的数量非常多，一些片段可能会剩余太小，无法容纳某些进程，从而形成外部碎片。② 与大页相比，小页需要更多的页表来管理内存空间，以识别每个页面的物理位置。

4.3 (1) D - Dirty D位为1，表明该页是否被改写。D位为0，对此页面写操作触发异常。

⇒ A - Accessed A位为1，该页可被访问，为0不可访问，否则触发异常。

⇒ G - Global 全局页面标识<sup>识</sup>，当前页可供多个进程共享。

U - User 用户模式可访问

X W R - 可执行，可写，可读

V - Valid 表明物理页在内存中是否分配好。

(2) 错误的操作将会导致严重的后果，极有可能导致系统~~崩溃~~进程的崩溃。

(3) 指向下一级页表。

4.4 (1) PMP控制寄存器的X/W/R位决定的是具体地址空间的读写与执行。

(2) A: 表项的地址匹配模式

L: 表项的Lock使能位

4.8 (1) 4KB  $\leftrightarrow$  ~~2~~<sup>12</sup>位  $64 - 12 = 52$ 位

~~2~~  $2^{52} \times 8 = 2^{55} = 2^{45}$  KB

(2)  $48 - 12 = 36$ 位  $2^{36} \times 8 = 2^{39} = 512$  G

(3) 多级页表相当于以页表为单位再创建更上一级的页表, 多级页表的优势在于能随着进程占用内存空间的增大, 对应地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。